



**TUGAS AKHIR - SM141501**

**IMPLEMENTASI *BAYESIAN NETWORK*  
UNTUK PERHITUNGAN PROBABILITAS  
PADA PENILAIAN RISIKO PIPA BAWAH  
LAUT OLEH FAKTOR KAPAL**

**FIRDA PUSPITA DEVI  
NRP 1212 100 090**

**Dosen Pembimbing  
Dr. Imam Mukhlash, S.Si, MT**

**JURUSAN MATEMATIKA  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**





**FINAL PROJECT - SM141501**

# **IMPLEMENTATION OF BAYESIAN NETWORK FOR PROBABILITY ON RISK ASSESSMENT OF OFFSHORE PIPELINE BY SHIPS FACTOR**

**FIRDA PUSPITA DEVI  
NRP 1212 100 090**

**Supervisor  
Dr. Imam Mukhlash, S.Si, MT**

**DEPARTMENT OF MATHEMATICS  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**



## LEMBAR PENGESAHAN

### IMPLEMENTASI BAYESIAN NETWORK UNTUK PERHITUNGAN PROBABILITAS PADA PENILAIAN RISIKO PIPA BAWAH LAUT OLEH FAKTOR KAPAL

### IMPLEMENTATION OF BAYESIAN NETWORK FOR PROBABILITY ON RISK ASSESSMENT OF OFFSHORE PIPELINE BY SHIPS FACTOR

#### TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat  
Memperoleh gelar Sarjana Sains  
Pada bidang minat Ilmu Komputer  
Program Studi S-1 Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember

Oleh :

**FIRDA PUSPITA DEVI**

NRP. 1212 100 090

Menyetujui,

Dosen Pembimbing,



Dr. Imam Mukhlash, S.SI. M.T

NIP. 19700831 199403 1 003

Mengetahui,  
Ketua Jurusan Matematika  
FMIPA-ITS



Dr. Imam Mukhlash, S.SI. M.T

NIP. 19700831 199403 1 003

Surabaya, 19 Januari 2017

# **IMPLEMENTASI BAYESIAN NETWORK UNTUK PERHITUNGAN PROBABILITAS PADA PENILAIAN RISIKO PIPA BAWAH LAUT OLEH FAKTOR KAPAL**

Nama Mahasiswa : Firda Puspita Devi  
NRP : 1212 100 090  
Jurusan : Matematika FMIPA-ITS  
Dosen Pembimbing : Dr. Imam Mukhlash, S.Si, MT

## **ABSTRAK**

Penyaluran minyak dan gas bumi dengan pipa bawah laut sangat efisien karena memerlukan biaya yang minimal. Namun, seiring padatnya aktivitas maritim oleh lalu lintas kapal akan berdampak kerusakan pada jaringan pipa bawah laut. Penyebab kerusakan tersebut dapat disebabkan oleh beberapa faktor, antara lain: jatuhnya jangkar kapal (*anchor drop*), kapal tenggelam akibat gagal mesin, jaring atau pukat (*trawl*), dan faktor lainnya. Oleh karena itu, penilaian risiko untuk kapal yang melewati jaringan pipa bawah laut sangat diperlukan. Pada penelitian ini, metode *Bayesian Network* (BN) digunakan untuk memodelkan penyebab kecelakaan pipa bawah laut oleh faktor kapal dan menghitung probabilitas kerusakan yang ditimbulkannya. Adapun standar DNV RP F107 digunakan untuk mengklasifikasi tingkat risiko berdasarkan nilai probabilitas yang didapat. Hasil pengujian program ini menunjukkan bahwa 58.4% kemungkinan kapal yang lewat tidak menyebabkan kerusakan pada pipa, 13.83% kerusakan yang ditimbulkan kecil, 15.14% kerusakan yang ditimbulkan menengah, dan 12.59% kerusakan yang ditimbulkan besar.

***Kata Kunci*** : *Pipa Bawah laut, Probabilitas, Bayesian Network, DNV RP F107.*

.



# **IMPLEMENTATION OF BAYESIAN NETWORK FOR PROBABILITY ON RISK ASSESSMENT OF OFFSHORE PIPELINE BY SHIPS FACTOR**

Name : Firda Puspita Devi  
NRP : 1212 100 090  
Department : Mathematics FMIPA-ITS  
Supervisor : Dr. Imam Mukhlash, S.Si, MT

## **ABSTRACT**

*The transportation of hydrocarbon by subsea pipeline is highly efficient while requiring minimal cost. However, with the rapid extension of offshore pipeline and the increasing of maritime activities, it can be reasonably expected that accident to offshore pipeline. Some hazards include anchor drop, drowing ship as a consequence to failed engine, trawl, etc. Hence the risk management is thus necessary with passing ship, which can be categorized as damage to offshore pipeline by “ship factors”. In this paper, Bayesian Network (BN) model are proposed to present a broad range of accident scenarios and determine final probabilities of anchor damage and trawling damage to subsea pipeline. In addition, DNV RP F107 is used to classify risk ranking according to the probability. As a result, 58.4% of ships passing isn't impact to pipeline damage, the 1<sup>st</sup> segment the percentage is 13.83% (low probability), the 2<sup>st</sup> segment the percentage is 15.14% (moderate), and the 3<sup>rd</sup> the percentage is 12.59% (major).*

**Keyword :** Subsea Pipeline, Probability, Bayesian Network, DNF RP F107.





## KATA PENGANTAR

Puji syukur penulis panjatkan pada kehadiran Allah Swt, karena hanya dengan karunia rahmat, bimbingan, serta anugrah-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul

### **“Implementasi *Bayesian Network* Untuk Perhitungan Probabilitas Pada Penilaian Risiko Pipa Bawah Laut Oleh Faktor Kapal”**

yang merupakan salah satu persyaratan akademik dalam menyelesaikan Program Sarjana Jurusan Matematika, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam proses pembuatan Tugas Akhir ini, penulis mendapat bantuan dan dukungan dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih kepada :

1. Ketua Jurusan Matematika FMIPA-ITS yang telah memberi dukungan dan kemudahan pengurusan persyaratan-persyaratan selama penulis menyelesaikan Tugas Akhir ini.
2. Bapak Dr. Imam Mukhlash, S.Si, MT sebagai dosen pembimbing Tugas Akhir atas segala bimbingan dan motivasi yang telah diberikan pada penulis.
3. Bapak Drs. I Gst Ngr Rai Usadha, M.Si sebagai dosen yang membantu penulis dalam pengerjaan Tugas Akhir ini.
4. Bapak Drs. Daryono Budi Utomo, MSi, Dr. Hariyanto, M.Si, Drs. Nurul Hidayat, M.Kom., Drs. Soetrisno, MI.Komp dan Ibu Alvida Mustika Rukmi, S.Si, M.Si selaku dosen penguji atas semua saran yang telah diberikan untuk perbaikan Tugas Akhir ini.
5. Kaprodi dan Bapak Drs. Iis Herisman, M.Sc selaku Sekretaris prodi S1 Jurusan Matematika ITS yang telah memberi dukungan dan kemudahan pengurusan persyaratan-persyaratan selama penulis menyelesaikan Tugas Akhir ini.

6. Bapak Drs. Soehardjoepri, M.Si selaku dosen wali penulis yang telah memberikan motivasi dan arahan akademik.
7. Bapak dan Ibu Dosen serta seluruh staf Tata Usaha dan Laboratorium Jurusan Matematika FMIPA-ITS.
8. Mas Danang dan Ikhwan selaku narasumber yang telah bersedia meluangkan waktu untuk memberikan informasi yang dibutuhkan penulis untuk menyelesaikan Tugas Akhir.
9. Keluarga tercinta terutama Bapak Mahfudz Sayudi dan Ibu Novaria Tuesdiana, penulis ucapkan banyak terima kasih atas doa serta dukungan yang telah diberikan baik moral maupun material, serta Khansa' Maisun Nabilah yang telah memberikan semangat dalam penyusunan Tugas Akhir ini.
10. Sahabat – sahabat penulis Lena, Sheerty, Maya, Laras, Alwi, Adit, dan Pipit yang telah memberikan semangat dan hiburan serta sebagai tempat berbagi apapun.
11. Teman – teman seperjuangan Matematika ITS 2012 khususnya MAT12IKS tercinta Haris, Hendra, Mia, Rere, Hakam, Sega, Hafiyyan, dan kawan-kawan yang telah banyak membantu baik secara langsung maupun tidak.
12. Sahabat penulis sedari SMA Fajar, Merry, Dayana, Reza, Audi, Syaiful, Sofi, dan Soca yang telah banyak memberikan saran dan motivasi.
13. Seluruh pihak yang tidak dapat penulis sebutkan satu persatu, yang turut membantu dalam penyusunan Tugas Akhir ini.

Penulis menyadari bahwa selama masa penelitian dan penyusunan laporan ini masih banyak kekurangan dan kesalahan. Oleh karena itu, penulis memohon saran dan kritik sebagai bahan perbaikan di masa yang akan datang. Semoga Tugas Akhir ini bermanfaat bagi semua pihak.

Surabaya, Januari 2017

Penulis

## DAFTAR ISI

JUDUL.....	i
LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
<i>ABSTRACT</i> .....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	5
1.6 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu.....	7
2.2 Identifikasi Bahaya ( <i>Hazard Identification</i> ).....	10
2.3 <i>Bayesian Network</i> (BN).....	12
2.4 Contoh Kasus dengan <i>Bayesian Network</i> .....	14
2.5 Standar (DNV) RP-F107.....	16
2.6 Sistem Pendukung Keputusan .....	18
2.7 Java.....	20
BAB III METODE PENELITIAN	23
3.1 Studi Literatur.....	23
3.2 Pengumpulan Data.....	23
3.3 Implementasi Sistem.....	23
3.4 Perancangan <i>Interface</i> dan Evaluasi Sistem.....	25

3.5	Penarikan Kesimpulan.....	26
3.6	Penulisan Laporan.....	26
BAB IV ANALISIS DAN PERANCANGAN SISTEM		29
4.1	Gambaran Umum Perangkat Lunak.....	29
4.2	<i>Usecase</i> Diagram.....	29
4.3	<i>Acitivity</i> Diagram.....	30
4.4	Membangun Sebuah <i>Network</i> atau Jaringan.....	31
4.5	Melakukan Inferensi dari Skenario.....	34
4.6	Simulasi Kasus.....	35
4.7	Menambahkan CPT.....	37
4.8	Desain <i>Interface</i> .....	38
BAB V HASIL DAN PEMBAHASAN PENGUJIAN		41
5.1	Deskripsi dari Pipa Bawah Laut dan Skenario Kejadian.....	41
5.2	Graf Kerusakan Akibat Jangkar dan Pukat.....	42
5.3	Mendapatkan <i>Marginal Probability Table</i> (MPT) dan <i>Conditional Probability Table</i> (CPT).....	44
5.4	Probabilitas Prediksi untuk Segmen Jalur Pipa.....	46
5.5	Analisis Permasalahan pada Kerusakan Jangkar.....	47
5.5	Analisis Permasalahan pada Kerusakan Pukat.....	50
BAB VI PENUTUP		
6.1	Kesimpulan.....	53
6.2	Saran.....	55
DAFTAR PUSTAKA.....		57
LAMPIRAN A CPT Akibat Kerusakan Jangkar.....		59
LAMPIRAN B Source Code.....		61
LAMPIRAN C Biodata Penulis.....		111

## DAFTAR GAMBAR

Gambar 1.1	Penyebab Kecelakaan pada Pipa Lepas Pantai.....	2
Gambar 2.1	Contoh Kasus dengan <i>Bayesian Network</i> ...	15
Gambar 2.2	Aplikasi DNV RP F-107.....	17
Gambar 2.3	Skema SPK.....	20
Gambar 3.1	Diagram Alir Metode Penelitian.....	27
Gambar 4.1	<i>Usecase</i> Diagram Sistem Aplikasi.....	30
Gambar 4.2	<i>Activity</i> Diagram Sistem Aplikasi <i>Bayesian Network</i> Untuk Perhitungan Probabilitas Pipa Bawah Laut oleh Faktor Kapal.....	31
Gambar 4.3	Membangun Sebuah <i>Network</i> .....	33
Gambar 4.4	Melakukan Inferensi.....	35
Gambar 4.5	Membaca <i>Network</i> dari File.....	36
Gambar 4.6	Membuat CPT.....	37
Gambar 4.7	Form Utama.....	38
Gambar 4.8	Form Pilihan Menu File.....	39
Gambar 4.9	Form Menu Open.....	40
Gambar 4.10	Form Menu Inferensi.....	40
Gambar 5.1	Graf Kerusakan oleh Jangkar.....	43
Gambar 5.2	Graf Kerusakan oleh Pukat.....	43
Gambar 5.3	MPT dari kedalaman menggunakan <i>Fuzzy</i> .....	46
Gambar 5.4	CPT <i>Anchor Damage</i> .....	48
Gambar 5.5	Hasil Inferensi Data Prior <i>Anchor Damage</i> ..	50



## DAFTAR TABEL

Tabel 2.1	Kemungkinan Hazard dari Faktor Luar.....	10
Tabel 2.2	Ranking Frekuensi DNV RP F-107.....	17
Tabel 5.1	CPT untuk <i>Ship Passing</i> dan <i>Anchor</i> .....	44
Tabel 5.3	Definisi Variabel Kerusakan Akibat Jangkar...	48
Tabel 5.4	Definisi Variabel Kerusakan Akibat Pukat.....	51





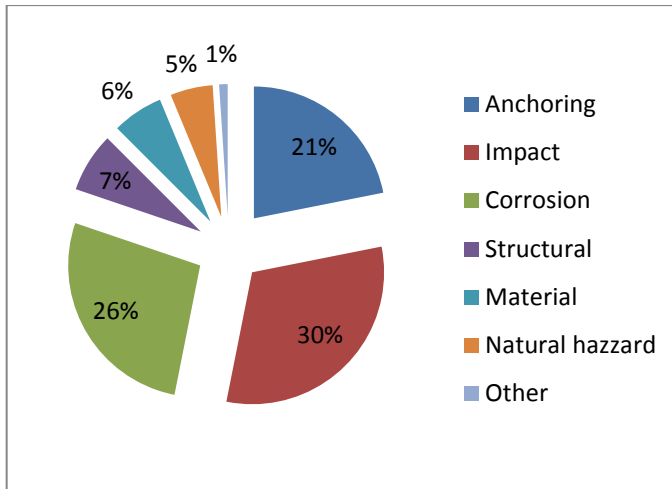
## **BAB I**

### **PENDAHULUAN**

Pada bab ini akan dijelaskan mengenai latar belakang dari permasalahan yang dibahas pada Tugas Akhir, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan dari Tugas Akhir ini.

#### **1.1 Latar Belakang Masalah**

Minyak dan gas bumi merupakan sumber daya alam yang menjadikan Indonesia sebagai negara dengan kapasitas produksi migas tertinggi di Asia Tenggara. Oleh karena itu, diperlukan mode transportasi untuk mengirimkan minyak dan gas bumi dari sumur produksi di laut ke pengolah di daratan. Demikian pula dari unit pengolah ke unit-unit distribusi. Penyaluran minyak dan gas bumi dengan pipa bawah laut diklaim sangat efisien karena memerlukan biaya yang minimal [1]. Akibatnya, pipa gas bawah laut menjadi pilihan utama untuk penyaluran minyak gas bumi di laut. Saat ini, Indonesia telah memiliki total panjang pipa bawah laut 9.340,72 km. Namun, dengan dicanangkannya tol laut oleh pemerintah Indonesia maka lalu lintas kapal akan semakin padat di atas pipa bawah laut. Dengan demikian terdapat bahaya atau *hazard* yang dapat ditimbulkan oleh faktor kapal (kapal tenggelam atau menjatuhkan jangkar pada saat *emergency*) yang berpotensi mengakibatkan kegagalan pengoperasian pipa gas bawah laut, yaitu kebocoran, pecah, atau bahkan ledakan. Hal ini dapat mengakibatkan gangguan transportasi, kerusakan pada lingkungan, dan bahkan gangguan kesehatan pada manusia. Data historis pada Gambar 1.1 menunjukkan bahwa sebagian besar kecelakaan pada pipa gas bawah laut disebabkan oleh dampak kecelakaan, korosi, dan jangkar kapal.



**Gambar 1.1** Penyebab Kecelakaan Pada Pipa Lepas Pantai [1]

Dari Gambar 1.1 dapat dilihat bahwa mayoritas kerusakan pipa gas bawah laut disebabkan oleh jangkar kapal dan dampak kecelakaan dari faktor kapal. Oleh karena itu, penilaian risiko untuk kapal yang melewati jaringan pipa gas bawah laut sangat diperlukan. Beberapa metode seperti algoritma *scoring-type* [2] dan beberapa metode kualitatif atau semi kuantitatif, yaitu AHP, logika fuzzy, dan *neural network* telah digunakan dalam model penilaian risiko pada jaringan pipa gas bawah laut [3]. Model penelitian risiko dengan menggunakan metode tersebut di atas memberikan hasil penelitian relatif yang mendukung penentuan tingkat risiko, namun belum dapat diterima oleh masyarakat. Saat ini penelitian telah bergeser dari sistem kualitatif menjadi kuantitatif probabilitas [4].

Metode *Fault Tree* (FT) telah terbukti efektif dalam menganalisis probabilitas kegagalan dan digunakan dalam teknik perpipaan[5,6]. Analisis FT adalah metode terbaik untuk memodelkan sistem sederhana yang statis. Namun, untuk sistem yang kompleks perlu menggunakan prosedur lainnya, seperti: *Bayesian Network*. *Bayesian Network* (BN) mempunyai struktur

yang jauh lebih fleksibel dan sesuai dengan berbagai macam kecelakaan daripada FT, sehingga dapat digunakan untuk menganalisis prosedur yang kompleks. Berdasarkan latar belakang tersebut, penulis ingin mengangkat topik tugas akhir dan diberi judul “Aplikasi *Bayesian Network* Untuk Perhitungan Probabilitas Pada Penilaian Risiko Pipa Bawah Laut Oleh Faktor Kapal”.

Penelitian ini mengacu pada salah satu standar, yaitu DNV RP-F107 (*Risk Assessment Of Pipelines Protection*). Standar DNV RP-F107 menyatakan bahwa bahaya (*hazard*) yang mungkin terjadi pada pipa gas bawah laut adalah bahaya-bahaya yang disebabkan oleh kejatuhan jangkar kapal (*anchor drop*), terseret jangkar (*anchor drag*), tertimpa kapal (*Ship Shinking*) [7]. Standar ini menyajikan pendekatan berbasis risiko atas kecelakaan akibat beban eksternal, merekomendasikan tindakan perlindungan, dan memberikan penilaian frekuensi dan konsekuensi untuk pipa gas bawah laut. Standar inilah yang nantinya menjadi tolok ukur pengambilan keputusan dalam penilaian risiko.

Berdasarkan Standard DNV RP-F107, risiko didefinisikan sebagai:  $R = P \times C$ , dengan R adalah nilai risiko, P adalah peluang/probabilitas timbulnya bahaya, dan C adalah konsekuensi dari bahaya tersebut. Bagian tersulit dari penilaian resiko adalah mengestimasi besarnya peluang bahaya yang akan muncul yang biasanya diperoleh dengan pemodelan atau simulasi. Pada tugas akhir ini akan dibangun suatu perangkat lunak yang mampu menghitung probabilitas kerusakan pada pipa gas bawah laut akibat faktor kapal yang lewat.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, permasalahan dalam Tugas Akhir ini dapat disusun sebagai berikut:

1. Bagaimana menganalisis struktur *Bayesian Network* yang merepresentasikan urutan suatu kejadian (*event*)

sehingga bahaya kecelakaan pada pipa gas bawah laut itu terjadi?

2. Bagaimana hasil implementasi metode *Bayesian Network* untuk menghitung probabilitas dari semua kejadian pada poin 1?

### 1.3 Batasan Masalah

Pada penelitian ini, permasalahan akan dibatasi pada hal-hal sebagai berikut:

1. Perangkat lunak dibangun menggunakan bahasa pemrograman Java.
2. Menggunakan standar dari *Det Norske Veritas* DNV RP-F107 “*Risk Assessment of Pipeline Protection*”.
3. Tidak melakukan penilaian risiko secara utuh pada obyek pengamatan (pipa bawah laut), melainkan hanya pada prosedur penghitungan probabilitas pada potensi bahaya (*hazard*).
4. Sumber potensi kerusakan yang diamati adalah potensi bahaya (*hazard*) yang terjadi setelah jalur pipa terpasang di dasar laut.

### 1.4 Tujuan

Berdasarkan permasalahan yang telah dirumuskan sebelumnya, tujuan penelitian Tugas Akhir ini adalah mengembangkan perangkat lunak yang dapat menghitung probabilitas penilaian risiko kegagalan pada pipa gas bawah laut dengan menggunakan metode *Bayesian Network* sehingga dapat diambil keputusan dalam tindakan preventifnya.

## 1.5 Manfaat

Setelah diperoleh probabilitas penilaian risiko kerusakan pada pipa gas bawah laut, maka Tugas Akhir ini dapat memberikan manfaat sebagai berikut :

1. Diperoleh hasil probabilitas pada kerusakan pipa gas bawah laut oleh faktor kapal menggunakan *Bayesian Network*.
2. Diperoleh tingkat risiko bahaya yang mungkin terjadi ketika pengoperasian pipa gas bawah laut akibat kapal yang lewat.
3. Diperoleh langkah mitigasi yang bertujuan untuk mengurangi risiko kerusakan pada pipa bawah laut.

## 1.6 Sistematika Penulisan

Sistematika dari penulisan penelitian ini adalah sebagai berikut:

### 1. BAB I PENDAHULUAN

Bab ini berisi tentang gambaran umum dari penulisan Tugas Akhir ini yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan.

### 2. BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang materi-materi yang mendukung Tugas Akhir ini, antara lain penelitian terdahulu, Identifikasi Bahaya (*Hazard Identification*), *Bayesian Network*, Standar (DNV) RP-F107, Sistem Pendukung Keputusan, serta penjelasan terkait aplikasi *Bayesian Network* Untuk Perhitungan Probabilitas Pada Penilaian Risiko Pipa Bawah Laut Oleh Faktor Kapal .

### 3. BAB III METODOLOGI PENELITIAN

Pada bab ini dibahas tentang langkah – langkah dan metode yang digunakan untuk menyelesaikan Tugas Akhir ini.

4. **BAB IV ANALISIS DAN PERANCANGAN SISTEM**

Pada bab ini akan menguraikan bagaimana tahapan tahapan dalam analisis dan perancangan sistem. Pembahasan analisis sistem dimulai dari deskripsi perangkat lunak hingga pemodelan analisis sistem, sedangkan perancangan sistem dimulai dari perancangan perangkat lunak hingga perancangan proses.

5. **BAB V HASIL DAN PEMBAHASAN PENGUJIAN**

Bab ini menjelaskan tentang implementasi rancangan perangkat lunak dalam bahasa pemrograman Java. Hasil uji coba perangkat lunak tersebut menggunakan data penelitian yang sudah ada.

6. **BAB VI PENUTUP**

Bab ini berisi kesimpulan yang diperoleh dari pembahasan masalah sebelumnya serta saran yang diberikan untuk pengembangan selanjutnya.

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini akan dijelaskan mengenai tinjauan pustaka yang menjadi dasar materi dalam penyusunan Tugas Akhir serta menunjang metode – metode yang digunakan dalam pembahasan Tugas Akhir ini.

#### **2.1 Penelitian Terdahulu**

Pipa gas bawah laut atau yang biasa disebut *offshore pipeline* memiliki potensi kerusakan yang jauh lebih besar daripada jalur pipa yang ada di darat. Dengan tren yang sedang berkembang belakangan ini mengenai *Health, Safety, dan Environment*, menjadikan alasan mengapa penelitian pada pipa bawah laut banyak dilakukan.

Fu Wang dkk dalam jurnal yang berjudul “Probability Analysis Of Offshore Fire By Incorporating and Organizational Factor”, menyatakan bahwa model analisis probabilitas pada pipa gas bawah laut direpresentasikan dengan mengubah metode *Fault Tree* (FT) menjadi *Bayesian Network* (BN) untuk menggabungkan akibat dari faktor manusia dan organisasi [14]. FT digunakan memodelkan faktor kejadian dalam penelitian tersebut, yang mana menjadi panduan dalam pembangunan struktur BN. Oleh karena itu, BN lebih sesuai untuk menghitung probabilitas dalam peristiwa yang belum pasti.

Penelitian penilaian risiko dengan menggunakan metode *Bayesian Network* disampaikan oleh Trucco dkk dalam jurnal yang berjudul “A Bayesian Network Modelling of Organisational Factors in Risk Analysis : A Case Study in Maritime Transportation”, menyatakan bahwa *Bayesian Network*



digunakan sebagai pendekatan yang memungkinkan penilaian probabilitas hubungan antara *basic event* dari suatu kecelakaan oleh transportasi laut dengan keadaan dan operasional dari suatu organisasi [11].

Topik mengenai penilaian risiko pada pipa gas bawah laut juga disampaikan dalam suatu paper konferensi tahunan, yaitu: *Transportation Research Board 92<sup>th</sup> Annual Meeting* [4]. Paper tersebut menggunakan metode *Bayesian Network* untuk menilai risiko kecelakaan pada pipa gas bawah laut oleh faktor kapal. Dalam paper tersebut dijelaskan bahwa BN digunakan dalam menentukan probabilitas kerusakan akibat jangkar dan pukut pada pipa bawah laut. Analisis probabilitas dinyatakan sebagai kunci utama dalam penilaian risiko dan pengambilan keputusan. Metode BN dalam paper tersebut dikembangkan dengan *directed acyclic graph* dan tiga metode perhitungan, antara lain: operasi *Boolean*, *standards* dan *historical data analysis*, dan teori *fuzzy set*.

Penelitian terkait dengan penilaian risiko pada pipa gas bawah laut diangkat oleh Yeyes M dengan judul *Estimation Method for Dragged Anchor Frequency on Subsea Pipelines*. Penelitian tersebut menggunakan metode *Bayesian Network* yang dikombinasikan dengan standar DNV dan Hanninen untuk memperkirakan frekuensi kejatuhan jangkar kapal pada daerah kritis, berdasarkan studi kasus di Selat Madura. Hasil dari penelitian tersebut adalah frekuensi jatuhnya jangkar per tahun kapal niaga masuk dalam kategori medium, yaitu level tiga ( $10^{-3} > x > 10^{-4}$ ) berdasarkan DNV RP F-107. Sedangkan untuk kapal angkatan laut, termasuk dalam peringkat frekuensi rendah, yaitu level dua (kurang dari  $10^{-4}$ ) [9]. Mitigasi dapat dilakukan di masa depan untuk mengurangi risiko kerusakan pada pipa bawah laut.

Dari penelitian tersebut di atas, didapat langkah-langkah dalam penilaian risiko yang terjadi pipa bawah laut adalah sebagai berikut :

1. Identifikasi bahaya (*Hazard identification*)

*Hazard Identification* adalah mengidentifikasi apa, bagaimana, dan mengapa suatu bahaya yang akan terjadinya, setelah itu akan dilakukan penilaian risiko untuk menghitung peluang kegagalan pipa akibat bahaya tersebut.

2. Perhitungan frekuensi dan konsekuensi

a. Perhitungan Frekuensi (Peluang Kejadian)

Perhitungan frekuensi dengan metode tertentu adalah untuk menghitung probabilitas kegagalan pipa, kemudian akan dianalisa banyaknya frekuensi yang akan terjadi pada setiap kejadian.

b. Perhitungan konsekuensi

Perhitungan konsekuensi dilakukan untuk mengetahui dampak atau akibat yang ditimbulkan dari kegagalan pipa.

3. Menentukan tingkat risiko

Menentukan tingkat risiko adalah tahap dimana dilakukan perkalian frekuensi dan konsekuensi yang dimasukkan ke dalam suatu matriks risiko. Untuk mengetahui matriks risiko maka digunakan pendekatan *standard code* atau standar acuan. Risiko tersebut akan ditentukan dalam *risk matrix* sehingga risiko dapat diterima atau tidak.

4. Mitigasi risiko

Pada tahap ini dilakukan evaluasi apakah risiko diterima atau tidak. Apabila risiko tersebut tidak dapat diterima, maka akan dilakukan proses mitigasi untuk menurunkan level risiko dengan mengurangi frekuensi dan konsekuensi yang terjadi.

## 2.2 Identifikasi Bahaya (*Hazard Identification*)

Bahaya atau *hazard* merupakan sumber potensi kerusakan atau situasi yang berpotensi menimbulkan kerugian. *Hazard Identification* adalah proses pengenalan terhadap bahaya yang mungkin terjadi, tanpa melihat besar dan kecilnya risiko yang diakibatkan dari suatu bahaya. Pengklasifikasian kategori bahaya yang mungkin terjadi pada pipa bawah laut digolongkan menjadi tiga kategori, yaitu [10] :

1. *Hazard* yang terjadi selama periode konstruksi
2. *Hazard* yang terjadi setelah jalur pipa terpasang di dasar laut
3. *Hazard* yang terjadi baik dari proses instalasi maupun dari proses operasi

*Hazard Identification* harus dapat melindungi semua kemungkinan terjadinya bahaya yang terjadi pada pipa gas bawah laut. Standard DNV RP F-107 telah memberikan contoh *hazard* yang mungkin dapat membahayakan jalur pipa gas bawah laut. Hal tersebut dapat dilihat pada Tabel 2.1 berikut :

**Tabel 2.1.** Kemungkinan *hazard* dari faktor luar

Operasi/ Aktivitas	<i>Hazard</i>	Konsekuensi yang Mungkin Pada Pipa
Pemasangan pipa	Pipa kejatuhan dan terseret jangkar kapal	Berdampak kerusakan
	Tumbukan akibat kapal menjatuhkan jangkar, dan lain-lain	
	Penurunan tekanan, penurunan pipa akhir, dan lain-lain	Kerusakan pada pipa/ <i>umbilical</i> terlempar pada pipa lain/ <i>umbilical</i>

		terpasang
	Kerusakan selama penggalian, pembuangan kerikil, pemasangan tutup pelindung, dan lain-lain	Berdampak kerusakan
	Kerusakan selama kontruksi penyebrangan	
Pemasangan anak tangga, ruangan, dan lain-lain (seperti pengangkat yang berat )	Kejatuhan benda	Berdampak kerusakan
	Kejatuhan rantai jangkar	Kerusakan, seperti tarikan dan goresan pada pipa
Perawatan jangkar (Melengkapi dan meletakkan kapal)	Kejatuhan jangkar, kerusakan akibat rantai jangkar	Berdampak kerusakan
	Kejatuhan jangkar	Tersangkut dan berdampak kerusakan
	Kejatuhan rantai jangkar	Kerusakan, seperti tarikan dan goresan pada pipa
Aktivitas pengangkatan (Melengkapi atau mengoperasikan platform)	Menjatuhkan objek di laut	Berdampak kerusakan
Operasi lepas pantai (Aktivitas yang bersamaan)	Dampak ROV	Berdampak kerusakan
	Kegagalan pengadaan selama proses instalasi peralatan/pemindahan	Berdampak kerusakan
		Kerusakan,

		seperti tarikan dan goresan pada pipa
Aktivitas nelayan	Dampak dari papan jaring/ kerusakan, seperti tarikan dan goresan pada pipa	Berdampak kerusakan dan tarikan
Lalu lintas kapal tanker, kapal suplai, dan kapal komersil	Tumbukan pada pipa (baik mesin kapal atau kapal kandas)	Berdampak kerusakan
	Penurunan jangkar darurat	Tersangkut dan berdampak kerusakan
	Kapal tenggelam (seperti setelah bertabrakan dengan platform atau kapal lain)	Berdampak kerusakan

### 2.3 *Bayesian Network* (BN)

*Bayesian Network* merupakan salah satu *Probabilistic Graphical Model* (PGM) sederhana yang dibangun dari teori probabilitas dan teori graf. Teori probabilistik berhubungan langsung dengan data, sedangkan teori graf berhubungan langsung dengan representasi yang ingin didapatkan [14]. Sebagai contoh, sebuah *Bayesian Network* dapat mewakili hubungan antara penyakit dan gejala. *Bayesian Network* dapat digunakan untuk menghitung probabilitas dari kehadiran berbagai gejala penyakit.

Metode *Bayesian Network* merupakan metode yang baik di dalam *machine learning* berdasarkan data *training*, dengan menggunakan probabilitas bersyarat sebagai dasarnya. *Bayesian Network* terdiri dari dua bagian utama, yaitu :

1. Struktur graf *Bayesian Network* disebut dengan *Directed Acyclic Graph* (DAG). DAG terdiri dari *node* dan *edge*.

*Node* merepresentasikan variabel acak dan *edge* merepresentasikan adanya hubungan ketergantungan langsung dan dapat juga diinterpretasikan sebagai pengaruh (sebab-akibat) antara variabel yang dihubungkannya. Tidak adanya *edge* menandakan adanya hubungan kebebasan kondisional di antara variabel.

## 2. Himpunan parameter

Himpunan parameter mendefinisikan distribusi probabilitas kondisional untuk setiap variabel. Pada *Bayesian Network*, *node* berkorespondensi dengan variabel acak. Tiap node diasosiasikan dengan sekumpulan peluang bersyarat,  $p(X_i|A_i)$  sehingga  $X_i$  adalah variabel yang diasosiasikan dengan *node* dan  $A_i$  adalah set dari *parent* dalam *graph*.

Dalam membangun *Bayesian Network*, struktur dibangun dengan pendekatan statistik yang dikenal dengan teorema *Bayes* yaitu *Conditional Probability* (peluang bersyarat). *Conditional Probability* yaitu perhitungan peluang suatu kejadian B bila diketahui kejadian A telah terjadi, dinotasikan dengan  $P(A)$ . Teorema ini digunakan untuk menghitung peluang suatu set data untuk masuk ke dalam suatu kelas tertentu berdasarkan inferensi data yang sudah ada. Adapun rumus dasar dari Theorema Bayes, yaitu:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

atau

$$P(A|B) = \frac{P(A)P(B|A)}{P(B|A)P(A) + P(B|\bar{A})P(\bar{A})} \quad (2)$$

*Bayesian Network* dapat melakukan pengambilan keputusan (inferensi) probabilistik. Inferensi probabilistik adalah memprediksi nilai variabel yang tidak dapat diketahui secara langsung dengan menggunakan nilai-nilai variabel lain yang telah

diketahui (Krause, 1998). Contoh inferensi probabilistik adalah menentukan probabilitas kondisional kerusakan akibat jatuhnya jangkar (*anchor damage*) jika diketahui dampak akibat kapal lewat dan dampak akibat beban jangkar terjadi. Inferensi probabilistik dapat dilakukan jika terlebih dahulu diperoleh *Joint Probability Distribution* (JPD) dari semua variabel yang dimodelkan. JPD adalah probabilitas semua kejadian variabel yang terjadi secara bersamaan.

Inferensi probabilistik dapat dilakukan jika struktur *Bayesian Network* telah dibangun. Dalam kasus penilaian risiko pipa bawah laut, hubungan antar variabel dan probabilitas nilai-nilai variabel belum diketahui. Adapun konstruksi *Bayesian Network* dari data terdiri dari dua tahap, yaitu :

1. Konstruksi struktur atau disebut juga tahap kualitatif, yaitu mencari keterhubungan antara variabel-variabel yang dimodelkan.
2. Estimasi parameter atau disebut juga tahap kuantitatif, yaitu menghitung nilai-nilai probabilitas.

## 2.4 Contoh Kasus dengan *Bayesian Network*

Misalkan diberikan lima variabel, yaitu: pencuri (*burglary*), gempa bumi (*earthquake*), alarm, John menelpon (*JohnCalls*), dan Mary menelpon (*MaryCalls*). Kita definisikan lima variabel di atas sebagai berikut:

B = Seorang pencuri masuk ke dalam rumah.

E = Gempa bumi yang terjadi di rumah

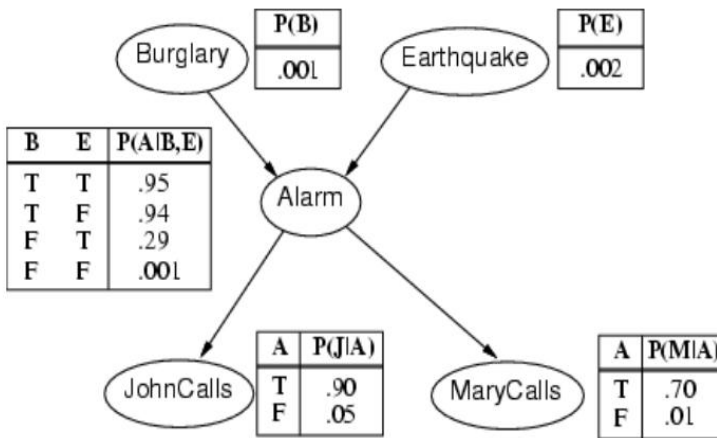
A = Alarm mati

J = John menelpon untuk melaporkan jika mendengar suara alarm

M = John menelpon untuk melaporkan jika mendengar suara alarm

Misal kita akan menghitung  $P(B|M, J)$ . Dengan menggunakan *joint probability*, maka akan dibangun Bayesian Network dengan langkah-langkah sebagai berikut:

1. Terdapat lima kejadian. Setiap kejadian mempunyai dua nilai kebenaran, yaitu benar dan salah, sehingga kemungkinannya adalah  $2^5 = 32$ .
2. Kita gunakan prior untuk setiap variabel Gambar 2.1.



**Gambar 2.1** Contoh Kasus dengan *Bayesian Network* [19]

3. Membangun Bayesian Network
  - a. Dari Gambar 2.1 dapat kita tuliskan partisi ordernya sebagai berikut:  
contoh :  
 $\{E, B\} \rightarrow \{A\} \rightarrow \{J, M\}$
  - b.  $P(J, M, A, E, B) = P(J, M|A, E, B) P(A|E, B) P(E, B)$   
 $\approx P(J, M|A). P(A|E, B).P(E).P(B)$   
 $\approx P(J|A).P(M|A).P(A|E, B).P(E) P(B)$

Ini adalah asumsi untuk *conditional independence* yang direpresentasikan dalam bentuk struktur graf dari *Bayesian Network*.



- c.  $P(J, M, A, E, B) = P(J|A) P(M|A) P(A|E, B) P(E) P(B)$
- d. Misal ada tiga *conditional probability table* yang akan ditentukan, yaitu:  $P(J|A)$ ,  $P(M|A)$ ,  $P(A|E, B)$ . Maka mempunyai kemungkinan  $2 + 2 + 4 = 8$ .
- e. Kemudian dua *marginal probabilities*  $P(E)$ ,  $P(B)$  mempunyai dua lagi kemungkinan
- f. Untuk menghitung  $P(B|M, J)$ , kita asumsikan  $P(b|m, \neg J)$  dimana  $P(B=True|M=True \wedge J = False)$ . Dari definisi, kita dapatkan

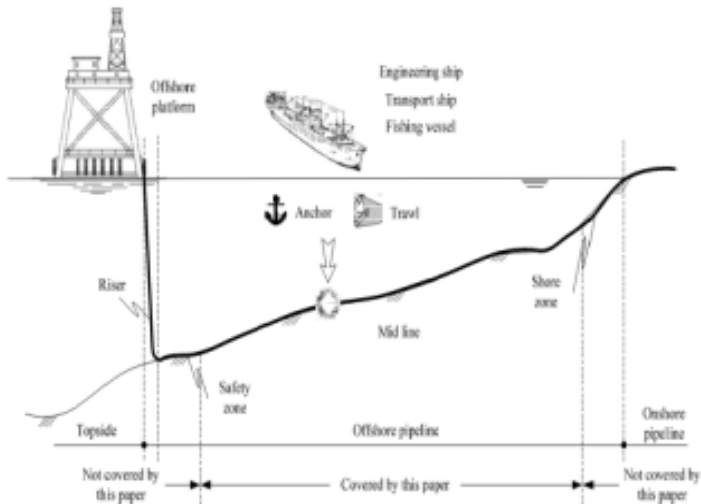
$$P(b|m, \neg j) = \frac{P(b, m, \neg j)}{P(m, \neg j)}$$

- g. Kemudian kita dapatkan *marginal probability* sebagai berikut:  
 $P(b|m, \neg j) = \sum_{A \in \{a, \neg a\}} \sum_{E \in \{e, \neg e\}} P(\neg j, m, A, E, B)$
- h. Dari *conditional independence* tersebut, kita dapat tuliskan sebagai berikut:  
 $P(J, M, A, E, B) \approx P(J|A) P(M|A) P(A|E, B) P(E) P(B)$   
 $P(\neg j, M, A, E, B) \approx P(\neg j|A) P(m|a) P(A|E, b) P(E) P(b)$ 
  - a. Untuk kasus  $A = a \wedge E = \neg e$   
 $P(\neg j, m, a, \neg e, b) \approx P(\neg j|A) P(m|a) P(A|E, b) P(E) P(b)$   
 $\approx 0.10 \times 0.70 \times 0.94 \times 0.998 \times 0.001$
- i. Begitu juga untuk kasus  $a \wedge e, \neg a \wedge e, \neg a \wedge \neg e$
- j. Sama dengan  $P(m, \neg j)$ , dapat kita partisi untuk mendapatkan  $P(b|m, \neg j)$

## 2.5 Standar (DNV) RP-F107

Dari beberapa standar penilaian risiko terhadap pipa gas bawah laut, DNV RP F107 (*Risk Assessment Of Pipeline Protection*) adalah salah satu standar yang paling sering digunakan. Standar ini hanya dapat diaplikasikan untuk pendekatan yang digunakan dalam melakukan penilaian risiko pada pipa gas bawah laut terhadap beban-beban eksternal, termasuk di dalamnya, antara lain: *risers, pipelines, umbilicals*,

dan proteksi yang perlu diberikan. Oleh karena itu, risiko yang muncul akibat korosi, erosi, kerusakan pipa akibat tekanan berlebihan tidak diatur dalam standar ini. DNV RP F-107 memiliki beberapa keterbatasan area penggunaan sebagaimana pada Gambar 2.2.



**Gambar 2.2** Aplikasi DNV RP F-107 [4]

Untuk membandingkan probabilitas kerusakan dan risiko dari bahaya yang berkaitan dengan pipa gas bawah laut, maka masing-masing peristiwa diberi peringkat dari 1 (probabilitasnya sangat rendah) sampai dengan 5 (probabilitasnya sangat tinggi) [8]. Sebagaimana disampaikan di atas, DNV RP F-107 membagi frekuensi menjadi lima tingkatan seperti Tabel 2.2.

**Tabel 2.2** Ranking frekuensi DNV RP F-107

Rangking	Deskripsi	Frekuensi Tahunan
1 (sangat rendah)	Frekuensi sangat rendah sehingga kejadian diabaikan	$< 1E-05$

2 (rendah)	Kejadian diperkirakan jarang terjadi	$1E-04 > 1E-05$
3 (medium)	Kejadian secara individu, diperkirakan tidak terjadi, tetapi dilihat dari numerik yang besar, kejadian dipercaya terjadi 1 kali selama setahun	$1E-03 > 1E-04$
4 (tinggi)	Kejadian secara individu mungkin terjadi satu kali selama usia pengoperasian pipa	$1E-02 > 1E-03$
5 (sangat tinggi)	Kejadian secara individu diperkirakan terjadi lebih dari satu kali selama usia pengoperasian pipa	$> 1E-02$

## 2.6 Sistem Pendukung Keputusan

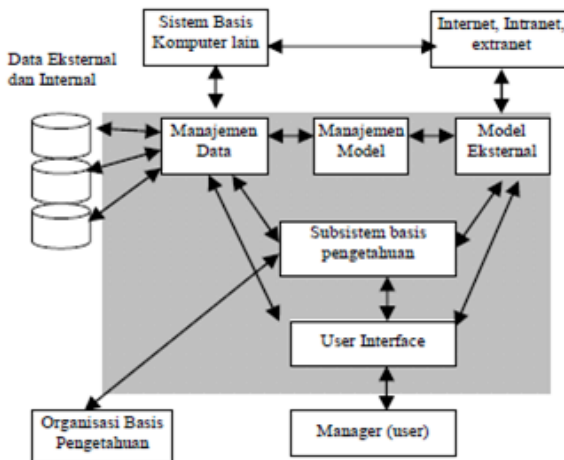
Sistem pendukung keputusan (SPK) adalah suatu sistem informasi spesifik yang ditujukan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur. Sistem ini memiliki fasilitas untuk menghasilkan berbagai alternatif yang secara interaktif dapat digunakan oleh pemakai. SPK terdiri dari beberapa komponen atau subsistem sebagai berikut.

### 1. Subsistem manajemen data

Subsistem manajemen data merupakan komponen SPK sebagai penyedia data bagi sistem. Data tersebut disimpan dalam suatu basis data yang diorganisasikan oleh suatu sistem yang disebut *database management system* (DBMS).

2. Subsistem manajemen model  
SPK mampu mengintegrasikan data dengan model-model keputusan. Model tersebut diorganisasikan oleh pengelola model yang disebut *model base*.
3. Subsistem antarmuka pengguna (*user interface*)  
Sistem diimplementasikan menggunakan *user interface* sehingga pengguna dapat berkomunikasi dengan sistem yang dirancang.
4. Subsistem manajemen pengetahuan  
Subsistem ini dapat mendukung subsistem yang lain atau bertindak sebagai suatu komponen independen. Subsistem ini juga memberikan inteligensi untuk memperbesar pengetahuan *decision maker*.

SPK harus mencakup tiga komponen utama yang terdiri dari DBMS, *model base*, dan *user interface*. Subsistem manajemen berbasis pengetahuan bersifat opsional, namun dapat memberikan banyak manfaat karena memberikan inteligensi bagi tiga komponen utama tersebut. Skema SPK dan komponen-komponennya ditunjukkan pada Gambar 2.3.



**Gambar 2.3** Skema SPK [18]

## 2.5 Java

Java adalah bahasa pemrograman yang dikembangkan oleh Sun Microsystems pada tahun 1991. Java merupakan bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi serta bersifat *open source*. Platform Java terdiri dari kumpulan library, *Java Virtual Machine* (JVM), sebuah *compiler*, *debugger*, dan perangkat lainnya yang dipaket dalam *Java Development Kit* (JDK). Aplikasi Java yang diciptakan dengan file teks berekstensi \*.java akan dikompilasi oleh JVM dan menghasilkan file *bytecode* berekstensi \*.class. Hal inilah yang membuat program Java dapat dijalankan pada berbagai platform sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

Program Java dapat dibedakan menjadi dua jenis, yaitu: applet dan aplikasi. Applet adalah program Java yang diletakkan

pada *Web server* dan diakses melalui *Web browser*. Dalam hal ini *browser* yang digunakan adalah yang memiliki kemampuan Java, misalnya: Internet Explorer. Sedangkan aplikasi adalah program Java yang bersifat umum. Aplikasi dapat dijalankan secara langsung, tidak perlu perangkat lunak *browser* untuk menjalankannya. Setelah dikompilasi, program dapat dieksekusi secara langsung [15].

Platform Java memiliki tiga edisi yang berbeda, yaitu:

1. J2SE (Java 2 Standard Edition)  
J2SE digunakan untuk mengembangkan aplikasi desktop dan applet. Java applet adalah program Java yang diakses melalui *Web browser* sehingga untuk menjalankan sebuah komputer harus memiliki program penjelajah web yang dapat menjalankan Java, seperti: Microsoft Internet Explore, Netscape Navigator, Mozilla firefox, dan Opera.
2. J2EE (Java 2 Enterprise Edition)  
J2EE adalah pengembangan Java yang ditunjukkan untuk sebuah perusahaan dengan skala cukup besar (*enterprise*). J2EEmengandung lebih banyak API (*Application Programming Interface*) serta arsitektur yang lebih besar daripada J2SE, J2EE, terdiri dari tiga bagian utama, yaitu: Client, Java EE Server, dan database.
3. J2ME (Java 2 Micro Edition)  
J2ME adalah pengembangan Java untuk perangkat mobile, seperti handphone, pocket PC, dan PDA [16].



## **BAB III**

### **METODE PENELITIAN**

Bab ini menguraikan metode yang digunakan dalam penelitian secara rinci. Metodologi penelitian yang digunakan berguna sebagai acuan sehingga penelitian dapat berjalan sistematis.

#### **3.1 Studi Literatur**

Pada tahap ini dilakukan pengumpulan teori pendukung mengenai dengan mencari referensi yang menunjang penelitian yang berupa tugas akhir, jurnal, buku, maupun artikel. Dalam Tugas Akhir ini, studi pustaka yang dilakukan mengenai penilaian risiko pipa gas bawah laut, *Bayesian Network*, dan DNV RP-F107.

#### **3.2 Pengumpulan Data**

Pengumpulan data merupakan tahap untuk mengumpulkan data yang diperlukan dalam pengerjaan Tugas Akhir ini. Pada Tugas Akhir ini, data diambil secara sekunder dari penelitian Transportation Research Board 92<sup>th</sup> Annual Meeting (2013) mencakup berbagai data mengenai CPT dan MPT.

#### **3.3 Implementasi Sistem**

Pada tahap ini dilakukan proses dimana persoalan diterjemahkan menjadi produk perangkat lunak melalui suatu rangkaian aktivitas sesuai model proses, metode, dan alat bantu yang akan digunakan. Tahap-tahap yang dilakukan adalah sebagai berikut :



a. *Hazard Identification*

Pada tahap ini dilakukan pencarian apa, mengapa, dan bagaimana penyebab kecelakaan pada pipa gas bawah laut yang digunakan sebagai dasar analisis selanjutnya.

b. Pengolahan data menggunakan metode *Bayesian Network*

Terdapat beberapa langkah dalam menerapkan metode *Bayesian Network*, sebagai berikut :

- i. Membangun struktur *Bayesian Network* dengan memperkirakan urutan-urutan suatu kejadian (*event*) sehingga bahaya kecelakaan pada pipa gas bawah laut itu terjadi.
- ii. Menentukan parameter (*Prior Probability*)  
Setelah struktur *Bayesian Network* terbentuk, langkah selanjutnya adalah menentukan parameter (*Prior Probability*) dari tiap-tiap penyebab kecelakaan pipa gas bawah laut. *Prior Probability* merupakan derajat kepercayaan dari suatu gejala yang digunakan ketika tidak ada informasi lain yang dapat digunakan untuk melihat kemungkinan suatu *event* terjadi, tetapi begitu informasi baru diketahui maka probabilitas harus diperbarui.
- iii. Membuat *Conditional Probability Table* (CPT)  
*Conditional Probability* adalah probabilitas suatu *event* B apabila *event* A telah terjadi. Setiap tabel yang berisi probabilitas dari setiap kemungkinan nilai A dan B disebut *Conditional Probability Table* (CPT).
- iv. Membuat *Joint Probability Distribution* (JPD)  
*Joint Probability Distribution* adalah probabilitas kemunculan bersama untuk semua kombinasi kemungkinan nilai-nilai yang terdapat pada variabel A dan B. Sama halnya dengan CPT, *Joint Probability Distribution* dari suatu variabel A dan B adalah sebuah tabel yang berisi probabilitas untuk setiap nilai A dan B yang dapat terjadi. Cara menghitung *Joint Probability*

*Distribution* suatu penyebab kecelakaan adalah dengan mengalikan nilai *Conditional Probability* dengan *Prior Probability*.

v. Menghitung *Posterior Probability*

*Posterior Probability* dapat dihitung dari hasil *Joint Probability Distribution* yang telah diperoleh yang kemudian digunakan untuk menghitung nilai probabilitas tiap-tiap penyebab kecelakaan.

vi. Inferensi Probabilistik

Data yang digunakan dalam melakukan inferensi diperoleh dari perhitungan frekuensi dengan penggabungan peluang frekuensi tahunan kapal melewati pipa, peluang kapal mengalami kondisi *emergency*, peluang jangkar jatuh pada daerah kritis (CADZ).

c. Menentukan tingkat risiko berdasarkan standar DNV RP-F107

Dari hasil inferensi probabilistik yang telah dihitung, maka didapatkan perkiraan frekuensi penurunan jangkar dari masing-masing kelompok kapal setiap segmen pipa. Kemudian rangking frekuensi kelompok kapal tersebut dikalsifikasikan sesuai dengan tabel frekuensi pada DNV RP F-107.

### 3.4 Perancangan Interface dan Evaluasi Sistem

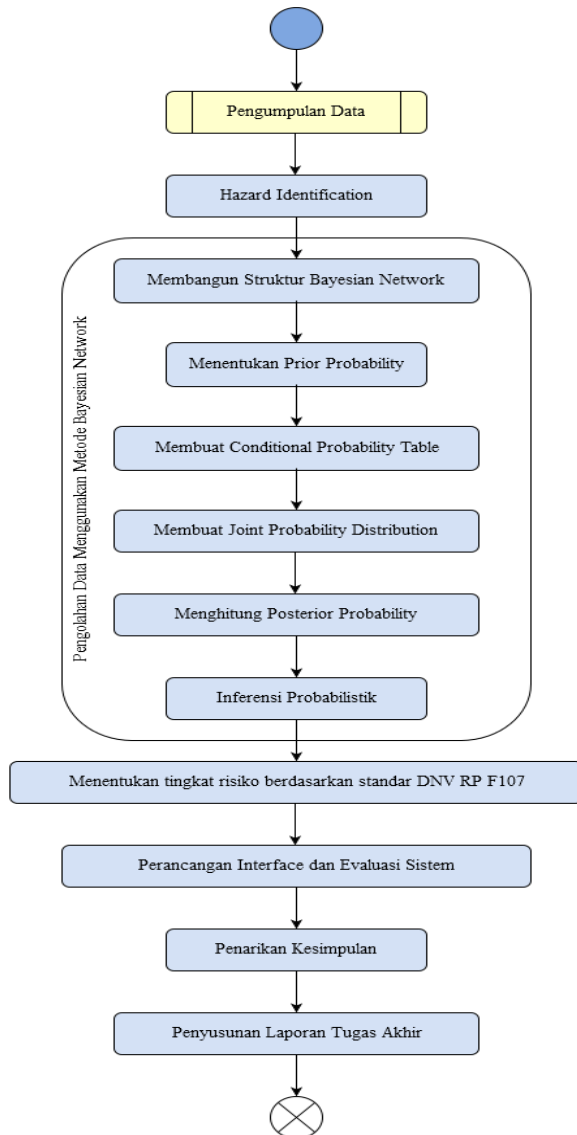
Sehubungan dengan pengolahan data, maka pada tahap implementasi adalah tentang bagaimana pengolahan datanya diterapkan dalam sebuah sistem. Sistem yang dibuat diimplementasikan dengan menggunakan bahasa pemrograman Java Netbeans. Penulis melakukan pengujian dan mengevaluasi sistem yang dihasilkan, serta melakukan perbaikan sistem. Selain itu juga melakukan pengujian terhadap sistem, apakah sudah sesuai dengan yang diharapkan

### **3.5 Penarikan Kesimpulan**

Setelah dilakukan analisis dan pembahasan maka dapat ditarik suatu kesimpulan dan saran sebagai masukan untuk pengembangan penelitian lebih lanjut. Setelah semua proses selesai dilakukan, maka tahap terakhir adalah penyusunan laporan tugas akhir.

### **3.6 Penulisan Laporan**

Pada tahap ini akan dilakukan penulisan laporan yang menjelaskan mengenai penelitian yang telah dilakukan berdasarkan sistematika penulisan buku tugas akhir yang ada.



**Gambar 3.1 Diagram Alir Metode Penelitian**



## **BAB IV**

### **ANALISIS DAN PERANCANGAN SISTEM**

Pada bab ini dijelaskan secara detail tentang spesifikasi kebutuhan perangkat lunak yang berupa informasi mengenai fungsi yang dibutuhkan untuk performansi dan *interface*. Hasil analisis kemudian didokumentasikan dalam bentuk desain yang berupa *usecase diagram*, *activity diagram*, rancangan dalam Java *library*, dan desain *interface* sehingga mudah dipahami.

#### **4.1 Gambaran Umum Perangkat Lunak**

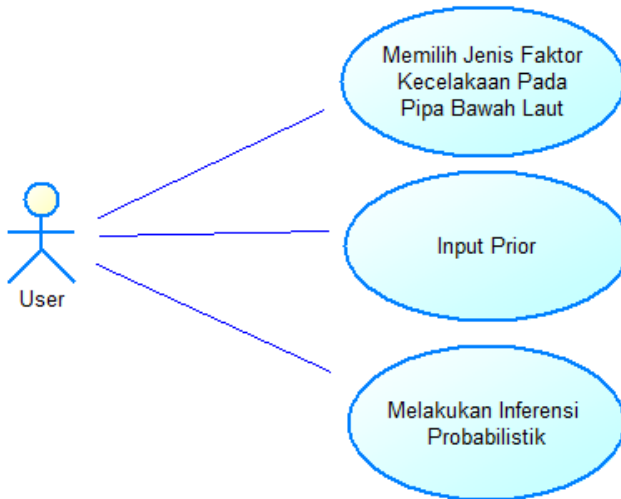
Secara keseluruhan program ini dibuat dengan menggunakan bahasa pemrograman Java. Adapun Java *library* atau perpustakaan yang digunakan adalah NeticaJ. NeticaJ adalah *library* yang memudahkan dalam penggunaan *Probabilistic Graphical Models* (PGM), khususnya *Bayesian Network* oleh para pemrogram untuk membangun aplikasinya.

Perangkat lunak ini berfungsi sebagai aplikasi yang memudahkan dalam perhitungan probabilitas dari suatu masalah menggunakan *Bayesian Network* sehingga *decision maker* dapat mengambil keputusan dengan memasukkan atribut kasus yang telah ditentukan. Perangkat lunak ini juga menyediakan fitur penyimpanan data hasil perhitungan.

#### **4.2 Usecase Diagram**

*Usecase diagram* menggambarkan interaksi apa saja yang dapat dilakukan di dalam sistem dan siapa saja aktor yang akan menjalankan interaksi tersebut. User dapat memilih struktur *graph* dari tipe *hazard* penyebab kecelakaan pada pipa bawah laut oleh faktor kapal, meng-inputkan prior probabilitas atau parameter, dan melakukan inferensi probabilistik atau skenario

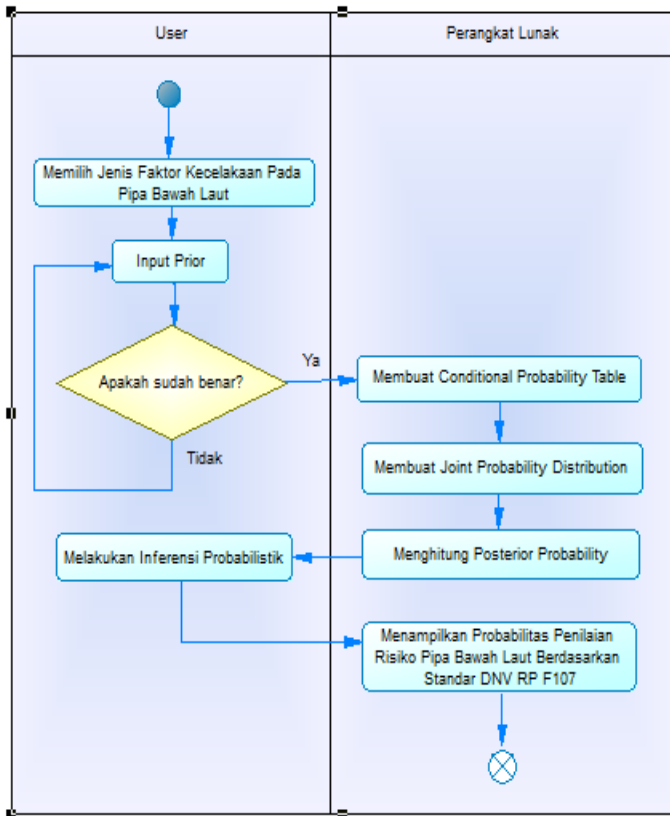
penyebab kecelakaan pipa bawah laut. *Use case diagram* tersebut dapat dilihat pada Gambar 4.1.



**Gambar 4.1** *Usecase Diagram* Sistem Aplikasi

### 4.3 Activity Diagram

Berdasarkan *use case diagram* yang ditunjukkan pada gambar 4.1, kemudian dibuat suatu *activity diagram* untuk menggambarkan alur proses dari setiap interaksi.

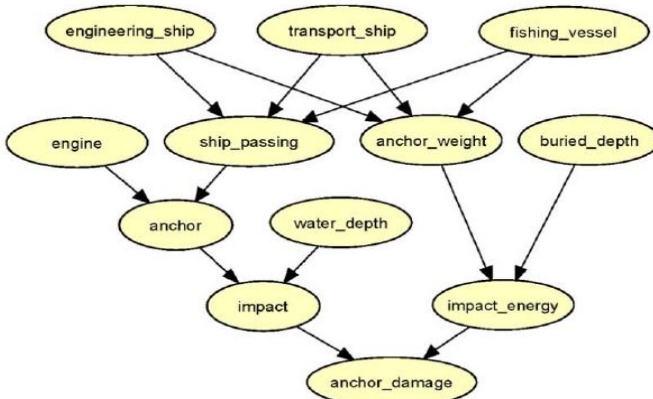


**Gambar 4.2** Activity Diagram Sistem Aplikasi Bayesian Network Untuk Perhitungan Probabilitas Pipa Bawah Laut Oleh Faktor Kapal

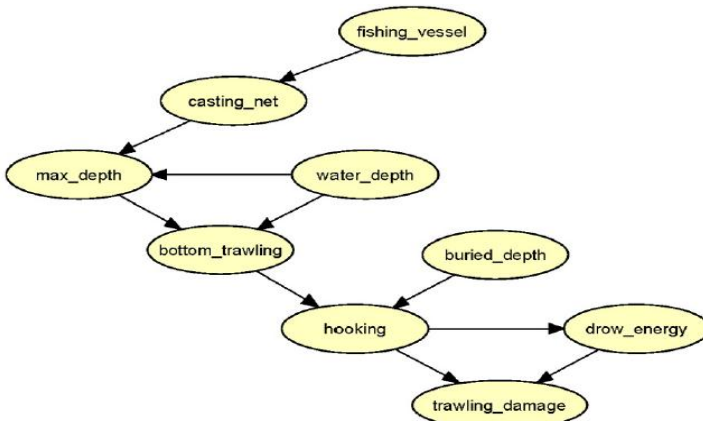
#### 4.4 Membangun Sebuah Network atau Jaringan

Pada proses pertama, kita asumsikan penyebab-penyebab kecelakaan pada oleh kapal yang melewati yang melewati pipa bawah laut. Kemudian dibangunlah sebuah jaringan *Bayesian Network* berdasarkan skenario di atas dengan graf asiklik pada gambar 4.3 dan 4.4 [4].





**Gambar 4.3** Graf Asiklik Berarah oleh Kerusakan Akibat Jangkar



**Gambar 4.4** Graf Asiklik Berarah Oleh Kerusakan Akibat Pukat

Proses dalam membangun sebuah network atau jaringan dapat dijelaskan dengan fungsi sebagai berikut:

```
private void buildGraph(String[] vertex, Point[] point, Point[]
edgePoint){
    graph.getModel().beginUpdate();
    try
    {
        for(int i = 0; i < vertex.length; i++){
            graph.insertVertex(parent, null, vertex[i],
            point[i].getX(), point[i].getY(),
            100,40,"shape=ellipse;perimeter=ellipsePeri
            meter");
        }
        for(int i = 0; i < edgePoint.length; i++){
            graph.insertEdge(parent, null, "",
            graph.getModel().getChildAt(parent,
            edgePoint[i].x),
            graph.getModel().getChildAt(parent,
            edgePoint[i].y));
        }
    }
    finally
    {
        graph.getModel().endUpdate();
    }
}
```

**Gambar 4.3** Membangun Sebuah *Network*

#### 4.5 Melakukan inferensi dari skenario

Setelah *network* dibangun, kita masukkan nilai dari setiap variabel ke dalam masing-masing node yang dinamakan *evidence*. Kemudian Netica melakukan inferensi probabilistik untuk menemukan kepercayaan (*beliefs*) dari variabel lain. Misal diberikan satu node yang terhubung dengan variabel “temperatur” lain, maka didefinisikan *value*-nya adalah dingin, medium, dan panas. Untuk temperatur dingin, medium, dan panas, masing-masing diketahui *belief* sebesar 0.1, 0.5, dan 0.4, menyatakan probabilitas dari temperatur dingin, medium, dan panas. Selanjutnya hasil akhir dari *belief* tersebut merupakan *posterior probability*.

Adapun proses dalam melakukan inferensi dapat dijelaskan dengan fungsi berikut:

```
Private void
inferenceBtnActionPerformed(java.awt.event.ActionEvent
evt) {
    try {
        net.compile();
        if(net.getTitle().equals("Anchor Damage")){
            resetInfPanel();
            anchorDamageInference ADI = new
            anchorDamageInference(net);
            inferencePanel.add(ADI);
            repaintInfPanel();
        }else if(net.getTitle().equals("Trawling Damage")){
            resetInfPanel();
            trawlingDamageInference TDI = new
            trawlingDamageInference(net);
            inferencePanel.add(TDI);
            repaintInfPanel();}
```

```

    }
    catch (NeticaException ex) {
        Logger.getLogger(BNSimple.class.getName()).log(Level.SEVERE, null, ex);
    }
}
catch (NeticaException ex) {
    Logger.getLogger(BNSimple.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

**Gambar 4.4** Melakukan Inferensi

## 4.6 Simulasi Kasus

Pada proses ini, program membaca network dari file dan menghasilkan suatu kasus sesuai yang kita inginkan beserta distribusi probabilitasnya. Proses tersebut dapat dijelaskan dijelaskan oleh fungsi sebagai berikut:

```

private void buildGraphFromFile(){
    try {
        NodeList nodes = net.getNodes();
        for(Object node:nodes){
            double[] position =
                net.getNode(node.toString()).visual().getPosition();
            graph.getModel().beginUpdate();

```

```

    try{
        graph.insertVertex(parent, null, node.toString(),
            position[0], position[1],
            100,40,"shape=ellipse;perimeter=ellipsePerimeter");
    }
    finally
    {
        graph.getModel().endUpdate()
    }
}
for(Object node:nodes){
    NodeList lp =
    net.getNode(node.toString()).getParents();
    Object chld = getvertex(node.toString());
    for(Object p:lp){
        Object prnt = getvertex(p.toString());
        graph.getModel().beginUpdate();
        try{
            graph.insertEdge(parent, null, "", prnt, chld);
        }
    finally
    {
        graph.getModel().endUpdate();
    }
}

} catch (NeticaException ex) {

    Logger.getLogger(BNSimple.class.getName()).log(Level.
    SEVERE, null, ex);

}

```

**Gambar 4.5** Membaca Network Dari File

#### 4.7 Menambahkan CPT

Pada proses ini, menambahkan CPT. Untuk setiap node, terdapat probabilitas dari setiap state yang bergantung pada setiap parent. Misal `cancer.setCPTable("smoker", 0.1, 0.9)` merupakan probabilitas jika kanker dalam *state* pertama yang mempunyai parent "smoker" adalah 0.1 dan probabilitas dalam *state* kedua adalah 0.9, sehingga dapat dinotasikan dengan probabilitas berikut:  $P(\text{cancer}=\text{true}|\text{smoking}=\text{true}) = 0.1$ . Proses tersebut dapat dijelaskan dengan fungsi sebagai berikut:

```
private void setCPT(String node, int row, int col) throws
NeticaException{
    int stateNum = net.getNode(node).getNumStates();
    float[] probs = new float[row*col];
    if(node.equals("engineering_ship")           ||
       node.equals("transport_ship")             ||
       node.equals("fishing_vessel")){
        for(int i = 0; i < row*col; i++){
            probs[i] = (float) (1.0/stateNum);
        }
    }else{
        for(int i = 0; i < row*col; i++){
            if(i%stateNum == 0){
                probs[i] = 1.0F;
            }else{
                probs[i] = 0.0F; }
        }
    }

    net.getNode(node).setCPTable(probs);
}
```

**Gambar 4.6** Membuat CPT

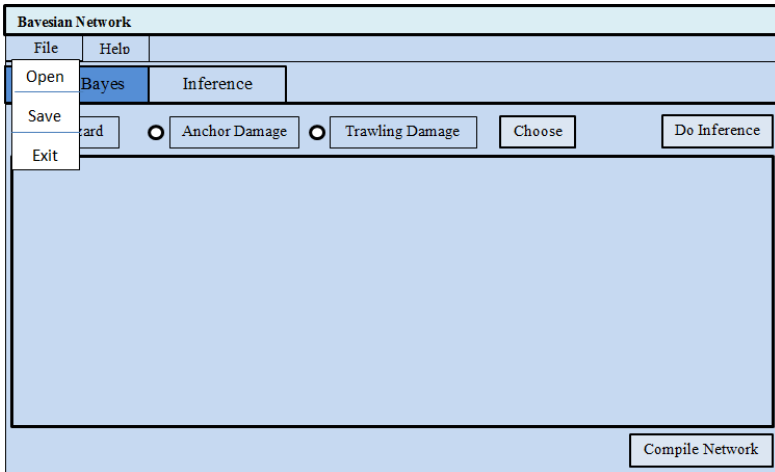
#### 4.8 Desain *Interface*

Subbab ini akan membahas mengenai *interface* awal program yang telah dibuat dengan menggunakan bahasa pemrograman Java. Gambar 4.7 merupakan desain tampilan awal yang muncul pada saat program dijalankan.

Bayesian Network				
File	Helo			
Struktur Bayes	Inference			
Tipe Hazard	<input type="radio"/> Anchor Damage	<input type="radio"/> Trawling Damage	Choose	Do Inference
Compile Network				

**Gambar 4.7** Form Utama

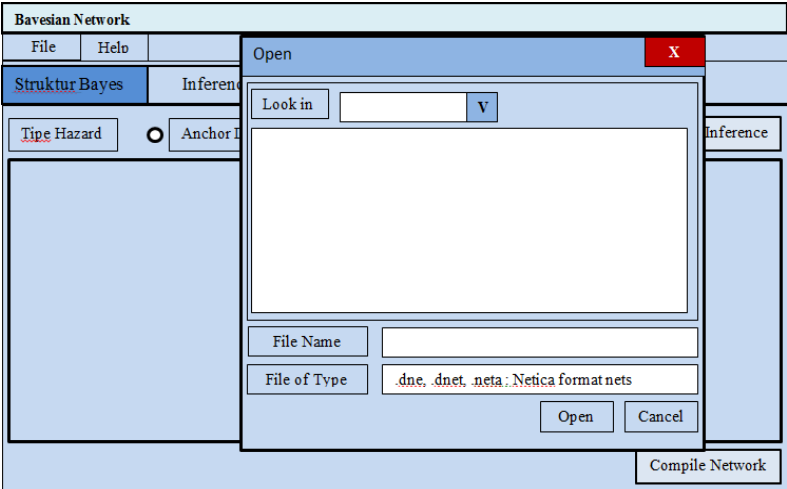
Setelah form utama, user dapat melihat beberapa menu yang terdapat pada pilihan menu bar di dalam form utama. Masing-masing menu bar tersebut memiliki menu item dan fungsi yang berbeda-beda. Pada menu bar file, terdapat tiga pilihan menu item yaitu:



**Gambar 4.8** Form Pilihan Menu File

Saat user mengklik menu open, maka akan muncul form seperti pada gambar 4.9. Pada form ini user dapat membuka data-data hasil perhitungan yang telah dilakukan sebelumnya. Begitu juga dengan menu save. Menu save dapat menyimpan data-data hasil perhitungan yang dapat dibuka kembali pada menu open.





Gambar 4.9 Form Menu Open

Setelah nilai CPT diperoleh, maka dilakukan inferensi. Desain halaman inferensi kemudian ditampilkan pada gambar 4.10 sebagai berikut.

Ship Passing	Engine Fail	Water Depth	Pipe Buried Depth
Engineering Ship <input type="radio"/> Passing <input type="radio"/> No <input type="radio"/> Don't Know	<input type="radio"/> Fail	<input type="radio"/> Shallow	<input type="radio"/> Shallow
Transport Ship <input type="radio"/> Passing <input type="radio"/> No <input type="radio"/> Don't Know	<input type="radio"/> Work	<input type="radio"/> Medium	<input type="radio"/> Medium
Fishing Vessel <input type="radio"/> Passing <input type="radio"/> No <input type="radio"/> Don't Know	<input type="radio"/> Don't	<input type="radio"/> Deep	<input type="radio"/> Deep
		<input type="radio"/> Don't Know	<input type="radio"/> Don't Know
<div>SubmitReset</div>			
<div>Ship Passing Probability</div> <div>Passing : Not Passing:</div>	<div>Impact Probability</div> <div>Impact : No:</div>	<div>Anchor Damage Probability</div> <div>No: Minor: Moderate: Major:</div>	
<div>Anchor Probability</div> <div>Passing : Not Passing:</div>	<div>Impact Energy Probability</div> <div>0 KJ : &lt;2,5 KJ : 2,5-10 KJ : 10-20 KJ : &gt;20 KJ :</div>		
<div>Anchor Weight Probability</div> <div>Passing : Not Passing:</div>			

Gambar 4.10 Form Menu Inferensi

## **BAB V**

### **HASIL DAN PEMBAHASAN PENGUJIAN**

Pada bab ini, dijelaskan tentang implementasi rancangan perangkat lunak dalam bahasa pemrograman Java. Adapun hasil uji coba perangkat lunak menggunakan data yang telah digunakan pada penelitian sebelumnya.

#### **5.1 Deskripsi Dari Pipa Bawah Laut dan Skenario Kejadian**

Kecelakaan akibat jangkar dan pukat (oleh faktor kapal) yang melewati jaringan pipa bawah laut sering terjadi dan telah dibahas lebih lanjut pada penelitian sebelumnya [17]. Penelitian tersebut menunjukkan bahwa frekuensi dari kerusakan pada pipa disebabkan oleh kejadian akibat jangkar dan pukat yang mempunyai persentase masing-masing sebesar 37% (19/52) dan 44% (22/52). Oleh karena itu, penelitian ini dibatasi dalam memodelkan kejadian akibat jangkar dan pukat saja.

Dalam menyusun model yang lengkap untuk perhitungan probabilitas skenario kejadian mempertimbangkan faktor-faktor sebagai berikut:

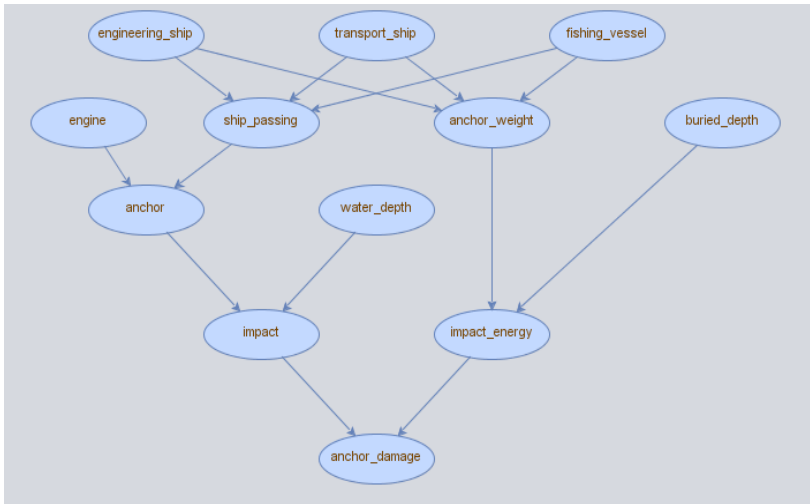
1. Kapal melewati pipa :  
Kapal teknik (*supply boat, crane ship*, dan sebagainya), kapal transport (*tanker, commercial ship, cruise ship*, dan sebagainya), dan kapal nelayan.
2. Aktivitas berisiko :  
Penurunan jangkar darurat (*anchor weight, anchor shape, bottom trawling*, tipe dari pukat pencari ikan (*type of fishing net*), kedalaman dari pelemparan pukat (*depth of casting net*), dan kekuatan dalam menarik pukat yang telah dilemparkan (*drow force*).
3. Karakteristik pipa bawah laut :  
(baja pelapis pipa, fleksibel atau terpusat), *water depth*, kedalaman pipa tertanam, fiameter, ketebalan dinding pipa, dan ketebalan pelapis pipa.

4. Kemungkinan konsekuensi pada pipa : *impact damage* dan *hooking damage*.

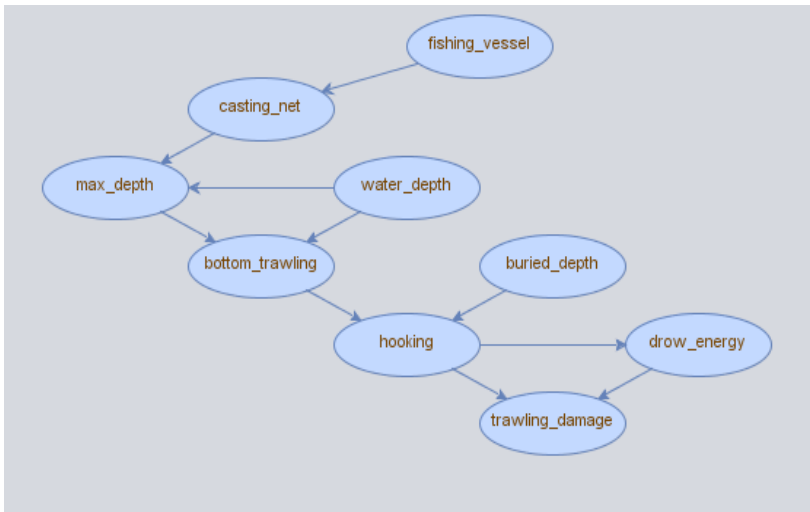
## 5.2 Graf Kerusakan Akibat Jangkar dan Pukat

Seperti yang telah ditampilkan pada gambar 5.1, kerusakan oleh jangkar didefinisikan sebagai faktor terjadinya kerusakan pada pipa bawah laut, dimana akibat dari jangkar terjadi ketika kapal lewat tepat diatas jalur pipa bawah laut, gagal mesin pada kapal, dan sebuah jangkar yang dijatuhkan dalam kondisi darurat. Dua faktor utama yang mengakibatkan efek dari energi adalah kedalaman air yang memberikan dampak sangat besar terhadap probabilitas dari jangkar dan lapisan pipa yang mampu mengurangi benturan terhadap pipa. Kedua dampak probabilitas dan dampak energi diintegrasikan untuk menentukan kerusakan yang diakibatkan jangkar terhadap pipa.

Seperti pada gambar 5.2, dampak pukat terhadap kerusakan pipa secara garis besar disebabkan oleh jaring yang menyangkut pada pipa. Kapal ikan mengeluarkan jaring sampai ke dasar laut, kemudian jaring tersebut tersangkut pada pipa bawah laut. Jaring yang tersangkut tersebut tetap ditarik oleh kapal sehingga tarikan tersebut akan merusak lapisan pada pipa. Tarikan dari jaring terhadap pipa merupakan sebuah fungsi dari *net-casting depth*, *water depth* dan *the pipeline coating*.



**Gambar 5. 1** Graf Kerusakan oleh Jangkar



**Gambar 5. 2** Graf Kerusakan Oleh Pukat

### 5.3 Mendapatkan *Marginal Probability Table (MPT)* dan *Conditional Probability Table (CPT)*

MPT dan CPT merupakan permasalahan yang cukup kompleks dan memerlukan pertimbangan yang cukup besar untuk secara penuh menghitung dan membentuk hubungan dalam bentuk model *Bayesian Network*. Dalam penelitian ini terdapat 3 langkah untuk mendapatkan CPT dan MPT.

Tabel 5.1 mengilustrasikan konversi dari operasi *boolean* ke dalam CPT, yang diterapkan dalam bentuk binari, dengan 2 kondisi yaitu 0 dan 1. Proses konversi tersebut dimulai dengan membandingkan *fault tree* dengan BN. Tabel 5.2 pada lampiran A menunjukkan contoh data dari CPT terhadap kerusakan oleh jangkar.

**Tabel 5. 1** CPT untuk *Ship Passing* dan *Anchor*

Operasi Boolean								
$ship\ passing = engineering\ ship \bigvee transport\ ship \bigvee fishing\ vessel$								
<i>Engineering ship</i>	<i>Passing</i>				<i>No</i>			
<i>Transport ship</i>	<i>Passing</i>		<i>No</i>		<i>passing</i>		<i>No</i>	
<i>Fishing vessel</i>	<i>passing</i>	<i>no</i>	<i>Passing</i>	<i>no</i>	<i>passing</i>	<i>no</i>	<i>passing</i>	<i>no</i>
<i>Ship passing</i>	1	1	1	1	1	1	1	0
<i>Passing =1, no = 0</i>	0	0	0	0	0	0	0	1
Operasi Boolean								
$anchor = ship\ passing \bigwedge engine$								
<i>Ship passing</i>	<i>Passing</i>				<i>No</i>			
<i>Engine</i>	<i>Fail</i>		<i>Work</i>		<i>Fail</i>		<i>Work</i>	
<i>Anchor</i>	1		0		0		0	
<i>Anchor=1, no=0</i>	0		1		1		1	

Keterangan untuk Tabel 5.2 pada lampiran A adalah sebagai berikut:

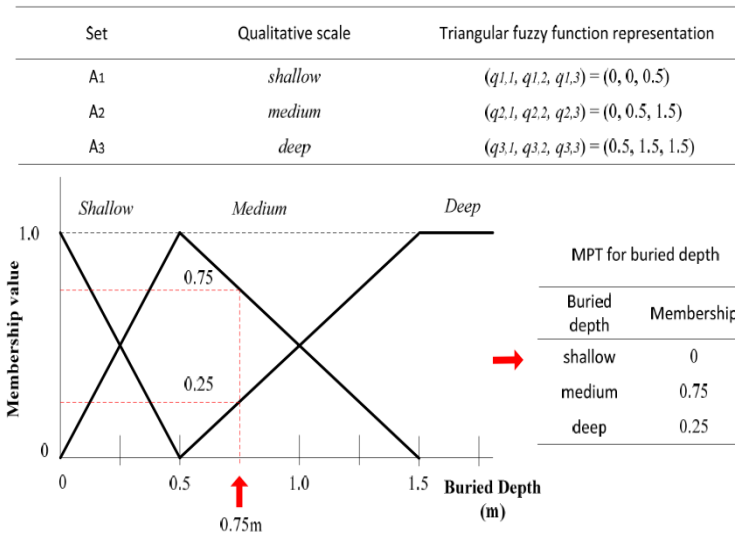
- a. *Minor Damage* :Kerusakan tidak memerlukan perbaikan atau mengakibatkan setiap pelepasan hidrokarbon.
- b. *Moderate Damage* :Kerusakan yang membutuhkan perbaikan, tetapi tidak mengarah untuk melepaskan hidrokarbon.
- c. *Major Damage* :Kerusakan yang mengarah untuk pelepasan hidrokarbon, air, dll

Akhirnya, penilaian ahli sangat penting untuk mendapatkan MPT dan CPT. Namun, para ahli biasanya lebih memilih untuk keputusan linguistik dari deskripsi probabilistik, seperti *safe* dan *unsafe*. Integrasi teori himpunan fuzzy dapat membantu domain expert untuk memperoleh MPT dan CPT dengan cara yang efisien. Misalnya, dalam tulisan ini kita mendefinisikan kedalaman terkubur dari pipa lepas pantai menggunakan tiga bilangan fuzzy.  $A_i$  didefinisikan dengan melalui semesta dimana setiap bagian mewakili kedalaman poin,  $A_1 = \textit{shallow}$ ,  $A_2 = \textit{medium}$ ,  $A_3 = \textit{deep}$ . Menurut pendapat para ahli, fungsi keanggotaan segitiga untuk setiap bilangan fuzzy diwakili oleh persamaan berikut [4].

$$\mu_{A_i}(x) = \begin{cases} 0, & 0 \leq x \leq q_{i,1} \\ \frac{x - q_{i,1}}{q_{i,2} - q_{i,1}}, & q_{i,1} \leq x \leq q_{i,2} \\ \frac{q_{i,3} - x}{q_{i,3} - q_{i,2}}, & q_{i,2} \leq x \leq q_{i,3} \\ 0, & x \geq q_{i,3} \end{cases} \quad (1)$$

X merupakan kedalaman yang akan dihitung nilai keanggotaannya.  $\mu_{A_i}(x)$  adalah nilai keanggotaan dari x terhadap  $A_i$ .  $q_{i,1}$  adalah kuartil atas dari  $A_i$ .  $q_{i,2}$  adalah kuartil tengah dari  $A_i$ . Dan  $q_{i,3}$  adalah kuartil bawah dari  $A_i$ .

Misalkan akan dihitung nilai keanggotaan pipa dengan kedalaman 0,75 meter, maka didapatkan bahwa  $\mu_{A_1}(x) = 0$ ,  $\mu_{A_2}(x) = 0.75$ , dan  $\mu_{A_3}(x) = 0.25$ . Himpunan fuzzy mewakili kedalaman terkubur dapat ditulis sebagai MPT, seperti ditunjukkan pada Gambar 5.3 [4].



**Gambar 5.3** MPT dari kedalaman menggunakan Fuzzy

Dengan menggunakan grafik pada Gambar 5.3, kita dapat menentukan secara langsung nilai keanggotaan dari suatu kedalaman pipa. Seperti pada Gambar 5.3, dengan mengambil kedalaman 0.75 meter, dapat diambil bahwa nilai keanggotaan untuk kedalaman medium adalah 0.75, dan nilai keanggotaan untuk kedalaman deep adalah 0.25. Sedangkan nilai keanggotaan untuk kedalaman shallow adalah nol dikarenakan dengan kedalaman 0.75 meter, garis yang merepresentasikan shallow tidak memiliki nilai.

## 5.4 Probabilitas Prediksi untuk Segmen Jalur Pipa

Prosedur untuk memperkirakan probabilitas kerusakan jangkar  $P^a$  digambarkan oleh persamaan :

$$P^a = \sum_{i=1}^n N_i \cdot P_i^a \quad (2)$$

Pada persamaan (2),  $i(1, \dots, n)$  adalah tipe kapal ke –  $i$ .  $N_i$  merupakan frekuensi tipe kapal ke –  $i$  yang melaluinya setiap tahun. Kemudian  $P_i^a$  dihitung dengan model BN menggambarkan probabilitas kerusakan jangkar sekali satu jenis kapal ke –  $i$  lewat. Pada Tugas Akhir ini, penulis hanya mempertimbangkan kapal teknik, kapal transportasi dan kapal nelayan. Adapun probabilitas kerusakan akibat jaring, sebagai berikut:

$$P^t = N_{fv} \cdot P_{fv}^t \quad (3)$$

di mana  $P^t$  adalah frekuensi kapal nelayan lewat tepat melalui jalur pipa lepas pantai per tahun, yang kemudian mempunyai kemungkinan kerusakan akibat pukot oleh salah satu kapal penangkap ikan.

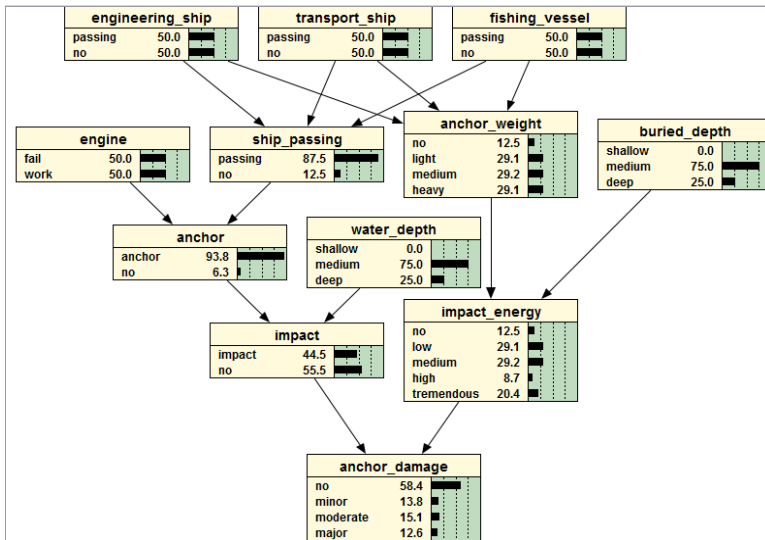
## 5.5 Analisis Permasalahan pada Kerusakan Jangkar

Pada subbab sebelumnya telah dibahas mengenai perhitungan untuk mendapatkan nilai MPT dan CPT. Dari subbab tersebut, kita dapat menentukan data *prior* kemudian akan diperoleh data *posterior*. Dengan menggunakan tools yang penulis buat, diinputkan data *prior* seperti gambar 5.4. Beberapa faktor yang mempengaruhi kerusakan jangkar diantaranya *engineering\_ship*, *transport\_ship*, *fishing\_vessel*, *ship\_passing*, *anchor\_weight*, *engine*, *buried\_depth*, *anchor*, *water\_depth*, *impact*, *impact\_energy*, *anchor\_damage*. Beberapa faktor tersebut akan didefinisikan pada tabel 5.3.



**Tabel 5. 2** Definisi Variabel Kerusakan Akibat Jangkar

<b>Variabel</b>	<b>Definisi</b>
<i>Engineering_ship</i>	Kapal Mesin
<i>Transport_ship</i>	Kapal penumpang
<i>Fishing_vessel</i>	Kapal penangkap ikan
<i>Ship_passing</i>	Kapal melalui daerah pipa gas bawah laut
<i>Anchor_weight</i>	Massa jangkar
<i>Engine</i>	Mesin kapal
<i>Buried_depth</i>	Kedalaman pipa gas ditanam
<i>Anchor</i>	Jangkar kapal
<i>Water_depth</i>	Kedalaman air
<i>Impact</i>	Dampak secara keseluruhan
<i>Impact_energy</i>	Dampak dari segi energi yang dilepaskan
<i>Anchor_damage</i>	Dampak dari kerusakan akibat jangkar kapal yang jatuh



**Gambar 5.4** CPT Anchor Damage

Berdasarkan CPT yang diperoleh dari gambar 5.4 maka akan dilakukan inferensi sehingga nantinya dapat diperoleh data *posterior*. Hasil dari inferensi kemudian ditampilkan pada gambar 5.5. Pada saat proses inferensi, kita dapat melakukan proses uji coba sehingga dapat diketahui seberapa besar faktor yang mempengaruhi kerusakan pada pipa gas.

Berdasarkan *running program* tersebut diperoleh hasil sebagai berikut:

1. Sebesar 87.5% kemungkinan kapal yang lewat akan berdampak kerusakan pada pipa bawah laut dan sebesar 12.5% kemungkinan kapal yang lewat tidak berdampak pada pipa.
2. Sebesar 93.75% kapal yang lewat akan menurunkan jangkar sehingga menyebabkan kerusakan pada pipa dan sebesar 6.25% tidak menurunkan jangkar.
3. Sebesar 12.50% kapal yang lewat tidak mengalami lego jangkar sehingga massa jangkar tidak berpengaruh pada

kerusakan pipa. Adapun kemungkinan kerusakan pipa akibat massa jangkar masing-masing sebesar 29.13%, 29.2%, dan 29.13% yang berada di tingkat *light*, *medium*, dan *heavy*.

4. Sebesar 44.5 % berdampak kerusakan pipa dan sebesar sisanya 55.47% tidak berdampak.
5. Sebesar 12.50% kerusakan pipa memiliki dampak energi yang dilepaskan berkisar 0 kJ. Kemudian sisanya, masing-masing sebesar 29.13%, 29.24%, 8.74%, dan 20.39% kerusakan tersebut memiliki dampak energi yang harus dilepaskan adalah berkisar kurang dari 2.5 kJ – 10 kJ, antara 2.5 kJ – 10 kJ, 10 – 20 kJ, dan lebih dari 20 kJ.
6. Sebesar 58.4% pipa tidak rusak akibat jangkar, 13.83% kerusakan yang ditimbulkan kecil, 15.14% kerusakan yang ditimbulkan menengah, dan 12.59% kerusakan yang ditimbulkan besar.

**Ship Passing**

Engineering Ship ☐ passing ☐ no ☒ don't know

Transport Ship ☐ passing ☐ no ☒ don't know

Fishing Vessel ☒ passing ☐ no ☒ don't know

**Engine Fail**

☐ fail

☐ work

☒ don't know

**Water Depth**

☐ shallow

☐ medium

☐ deep

☒ don't know

**Pipe Buried Depth**

☐ shallow

☐ medium

☐ deep

☒ don't know

**Ship Passing Probability**

Passing : 87.50%

Not passing : 12.50%

**Impact Probability**

Impact : 44.53125%

No : 55.46875%

**Anchor Damage Probability**

No : 58.437496%

Minor : 13.834375%

Moderate : 15.138844%

Major : 12.589282%

**Anchoring Probability**

Anchoring : 93.75%

Not anchoring : 6.25%

**Impact Energy Probability**

0 kJ : 12.50%

< 2.5 kJ : 29.124999%

2.5 - 10 kJ : 29.249999%

10 - 20 kJ : 8.7375%

> 20 kJ : 20.387501%

**Anchor Weight Probability**

No anchor : 12.50%

Light anchor : 29.124999%

Medium anchor : 29.249999%

Heavy anchor : 29.124999%

**Gambar 5. 5 Hasil Inferensi Data Prior Anchor Damage**

## 5.6 Analisis Permasalahan pada Kerusakan Pukat

Pada subbab sebelumnya, sudah dijelaskan mengenai permasalahan kerusakan yang disebabkan oleh jangkar. Oleh karena itu, subbab ini hanya akan menjelaskan mengenai faktor apa saja yang mempengaruhi kerusakan akibat pukot terhadap pipa gas. Beberapa faktor yang mempengaruhi diantaranya, *fishing\_vessel*, *casting\_net*, *max\_depth*, *water\_depth*, *bottom\_trawling*, *buried\_depth*, *hooking*, *drow\_energy* dan *trawling\_damage*. Beberapa faktor tersebut akan didefinisikan pada tabel 5.4.

**Tabel 5. 3** Definisi Variabel Kerusakan Akibat Pukat

<b>Variabel</b>	<b>Definisi</b>
<i>Fishing_vessel</i>	Kapal penangkap ikan
<i>Casting_net</i>	Jala
<i>Max_depth</i>	Kedalaman maximum
<i>Water_depth</i>	Kedalaman air
<i>Bottom_trawling</i>	Pukat bagian dasar
<i>Buried_depth</i>	Kedalaman penanaman pipa gas
<i>Hooking</i>	Pengait
<i>Drow_energy</i>	Energi untuk menenggelamkan
<i>Trawling_damage</i>	Kerusakan yang diakibatkan pukot

## 5.7 Mitigasi Risiko

Dari hasil pengujian di atas, maka perlu dilakukan langkah mitigasi yang bertujuan untuk mengurangi risiko yang ada. Karena pada studi kasus ini tidak mempertimbangkan masalah biaya, maka beberapa hal yang perlu dilakukan adalah :

1. Memberikan proteksi tambahan pada pipa jika mungkin terjadi bahaya kejatuhan jangkar dan tersangkut pukot. Proteksi tambahan yang mungkin dapat dilakukan adalah dengan menambahkan *box culvert* di sepanjang jalur pipa yang dilewati oleh jalur pelur pelayaran kapal.

2. Hal yang mungkin dilakukan pada pipa bawah laut adalah dengan menanam pipa dengan kedalaman tertentu di dasar laut. Kedalaman yang diperlukan untuk mengetahui tingkat keamanan pipa jika mungkin terjadi skenario bahaya kejatuhan jangkar dan terseret pukat, perlu dilakukan analisa dan perhitungan lebih lanjut.
3. Hal yang mungkin juga dilakukan adalah dengan mengurangi intensitas operasi kapal yang melewati jalur pipa bawah laut pada saat kondisi cuaca sedang buruk. Hal ini berfungsi untuk mengurangi frekuensi kemungkinan kapal menjatuhkan jangkar dalam situasi darurat.
4. Memberikan tanda di daerah jalur pipa tersebut yang menunjukkan informasi bahwa di daerah tersebut terdapat jalur pipa bawah laut, sehingga awak kapal akan mengerti bahwa ada jalur pipa yang membentang yang sedang dilewati oleh kapal.

## **BAB VI PENUTUP**

Bab ini berisi tentang kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan serta saran yang diberikan untuk pengembangan penelitian selanjutnya.

### **6.1 Kesimpulan**

Berdasarkan analisis terhadap hasil pengujian program, maka dapat diambil kesimpulan sebagai berikut :

1. Hasil pengujian berdasarkan nilai *prior*/parameter yang telah ditentukan oleh para ahli, menunjukkan bahwa perangkat lunak yang dibangun memberikan hasil nilai probabilitas beserta tingkat risiko kecelakaan pada pipa sebagai berikut:
  - a. Sebesar 87.5% kemungkinan kapal yang lewat akan berdampak kerusakan pada pipa bawah laut dan sebesar 12.5% kemungkinan kapal yang lewat tidak berdampak pada pipa.
  - b. Sebesar 93.75% kapal yang lewat akan menurunkan jangkar sehingga menyebabkan kerusakan pada pipa dan sebesar 6.25% tidak menurunkan jangkar
  - c. Sebesar 12.50% kapal yang lewat tidak mengalami lego jangkar sehingga massa jangkar tidak berpengaruh pada kerusakan pipa. Adapun kemungkinan kerusakan pipa akibat massa jangkar masing-masing sebesar 29.13%, 29.2%, dan 29.13% yang berada di tingkat *light*, *medium*, dan *heavy*.
  - d. Sebesar 44.5 % berdampak kerusakan pipa dan sebesar sisanya 55.47% tidak berdampak.
  - e. Sebesar 12.50% kerusakan pipa memiliki dampak energi yang dilepaskan berkisar 0 kJ. Kemudian sisanya, masing- masing sebesar 29.13%, 29.24%, 8.74%, dan 20.39% kerusakan tersebut memiliki dampak energi yang

- harus dilepaskan adalah berkisar kurang dari 2.5 kJ – 10 kJ, antara 2.5 kJ – 10 kJ, 10 – 20 kJ, dan lebih dari 20 kJ.
- f. Sebesar 58.4% pipa tidak rusak akibat jangkar, 13.83% kerusakan yang ditimbulkan kecil, 15.14% kerusakan yang ditimbulkan menengah, dan 12.59% kerusakan yang ditimbulkan besar.
  2. Dari kesimpulan yang di atas, maka perlu dilakukan langkah mitigasi, antara lain: memberikan proteksi tambahan pada pipa, menanam pipa dengan kedalaman tertentu, mengurangi intensitas operasi kapal yang melewati jalur pipa bawah laut pada saat kondisi cuaca seang buruk, dan memberi tanda di daerah jalur pipa.

## 6.2 Saran

Adapun saran dari Tugas Akhir ini adalah untuk pengembangan perangkat lunak serupa diperlukan metode terkait, seperti:

1. Dari tools ini dapat dikembangkan menjadi aplikasi yang dapat melakukan pemrosesan *Bayesian Network* secara umum
2. Dari hasil mitigasi dapat dilakukan penelitian lebih lanjut.

## DAFTAR PUSTAKA

- [1] Liu, Y. T., H. Hu, dan Y. B. Song. (2011) **“Pipeline Integrity Management System Based on Dynamic Risk Assessment”**. Journal of Shanghai Jiaotong University, Vol.45, No.5. (pp 687-690).
- [2] Muhlbauer, W. K. (2003). **Pipeline Risk Assessment Manual (Third Edition)**. Publication PARLOC 2001. Mott MacDonald Ltd., Offshore Operators Association, U.K. and the Institute of Petroleum.
- [3] Doremami, N., A. Afshar, A. D. Mohammadi. (2010). **“Hierarchical Risk Assessment in Gas Pipelines Based on Fuzzy Aggregation”**. Proceedings of 2<sup>nd</sup> International Conference on Reliability, Safety & Hazard.
- [4] Liu, Yutao. (2013). **“Probability Analysis of Damage to Offshore Pipeline by Ship Factors”**. Transportation Research Board 92<sup>th</sup> Annual Meeting.
- [5] Dong, Y., dan D. T. Yu. (2005). **“Estimation of Failure Probability of Oil and Gas Transmission Pipelines by Fuzzy Fault Tree Analysis”**. Journal of Loss Prevention in the Process Industries. Vol.18. (pp.83-88).
- [6] Wang Q., dan J. P. Zhao. (2008). **“Fuzzy Fault Tree Analysis Method on Submarine Pipelines Fault by The Third-Part Damage”**. *Nature Gas Indutry*, Vol.28. No.3. (pp.109-111).
- [7] **Risk Assessment of Pipelines Protection**. (2010). Norway. Publication DNV-RP-F107. Det Norske Veritas (DNV).
- [8] Sun, D, J., L. (2010). Elefteriadou. **“Research and implementation of lane-changing model based on driver behavior”**. In Journal of Transportation Research Board: Transportation Research Record 2161. (pp. 1-10).
- [9] Mulyadi, Y., Khanfir, S., Pitana, T., dkk. (2008). **“Estimation Method for Dragged Anchor Frequency on Subsea Pipelines in Busy Port Areas”**. Surabaya.



- [10] Mouselli, AH. (1981). **“Offshore Pipelines Design, Analysis and Methods”**. Oklahoma. PennWellBooks.
- [11] Trucco P., Cagno P., Ruggeri F., dkk. (2007). **“A Bayesian Network Modelling of Organisational Factors in Risk Analysis : A Case Study in Maritime Transportation”**. Milan, Italy. ScienceDirect. (pp. 823-834).
- [12] Turban, E., Aronson, J. E., & Liang, T.-P. (2005). **Decision Support Systems and Intelligent Systems (Sistem Pendukung Keputusan dan Sistem Cerdas)**. Yogyakarta: ANDI.
- [13] Wang, Y. F., M. Xie, K. M. Ng, Habibullah M S. (2011). **Probability Analysis of Offshore Fire by Incorporating Human and Organizational Factor**. *Ocean Engineering*. Vol. 38. (pp 2042-2055).
- [14] Heckerman, D., (1986). **Probabilistic Interpretations for Mycin’s Certainty Factors**. Elsevier Science Publishers B. V., North Holland.
- [15] Kadir, A. (2005). **Dasar Pemrograman Java 2**. Yogyakarta: ANDI.
- [16] Hariyanto, B. (2011). **Esensi-Esensi Bahasa Pemrograman Java**. Bandung: Informatika.
- [17] Mott MacDonald Ltd., Offshore Operators Association, U.K. and The Institute Of Petroleum. (2003). **The Update Of The Loss Containment Data For Offshore Pipeline**. U.K. Publication PARLOC 2001.
- [18] Turban, E., Aronson, J. E., & Liang, T.-P. (2005). **Decision Support Systems and Intelligent Systems (Sistem Pendukung Keputusan dan Sistem Cerdas)**. Yoyakarta: ANDI.
- [19] K. Murphy. **Dynamic Bayesian Network: Representation, Inference and Learning**. PhD thesis, UC Berkeley. 2002

## LAMPIRAN A

**Tabel 5.2** CPT Akibat Kerusakan Jangkar

		no		low		Medium		High		tremendous	
<i>Impact energy</i>		0 kJ		<2.5 kJ		2.5 – 10 kJ		10 – 20 kJ		>20 kJ	
<i>Impact</i>		impact	No	Impact	No	Impact	No	Impact	No	Impact	No
<i>Damage</i>	No	1	1	0	1	0	1	0	1	0	1
	Minor	0	0	1	0	0	0	0	0	0	0
	Moderate	0	0	0	0	0.5	0	0.25	0	0	0
	Major	0	0	0	0	0.5	0	0.75	0	1	0

## **LAMPIRAN B**

### *Source Code*

#### 1. Kelas Utama

```
public class BNSimple extends javax.swing.JFrame {

    private mxGraph graph;
    private Object parent;
    private Net net;
    private mxGraphComponent graphComponent;
    Environ env;

    public BNSimple() {
        initComponents();
        this.setTitle("Bayesian Network");
        jTabbedPane1.setEnabledAt(1, false);
        chooseBtn.setEnabled(false);
        compileBtn.setEnabled(false);
        inferenceBtn.setEnabled(false);
        Dimension dim =
Toolkit.getDefaultToolkit().getScreenSize();
        this.setLocation(dim.width/2-this.getSize().width/2,
dim.height/2-this.getSize().height/2);
        try {
            env = new Environ (null);
        } catch (NeticaException ex) {
            Logger.getLogger(BNSimple.class.getName()).log(Level.SEVERE,
            null, ex); }
    }
```

```

graph = new mxGraph();
    parent = graph.getDefaultParent();
    graphComponent = new mxGraphComponent(graph);

graphComponent.getGraphControl().addMouseListener(new
MouseListener() {

    @Override
    public void mouseClicked(MouseEvent e) {
        if(e.getButton() == 3){
            Object cell =
graphComponent.getCellAt(e.getX(), e.getY());
            if (cell != null && !graph.getModel().isEdge(cell))
            {
                JPopupMenu pop = new JPopupMenu();
                JMenuItem it1 = new JMenuItem("Conditional
Probability Table");
                it1.addActionListener(new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent e)
{
                        try {
                            displayCPT(graph.getLabel(cell));
                        } catch (NeticaException ex) {

Logger.getLogger(BNSimple.class.getName()).log(Level.SEVERE,
null, ex);    }    }

```

```

private void displayCPT(String label) throws NeticaException {
    float[] probs = net.getNode(label).getCPTTable(null);
    Point dim = getCPTSize(label);
    String[] comb = getComb(label);
    String[] state =
getColHeader(net.getNode(label).getNumStates(), label);
    Object[][] data = getCPdata(dim.x, dim.y, probs,
comb);
    JTable CPT = new JTable(data, state){
        @Override
        public boolean isCellEditable(int row, int column) {
            return column == 0 ? false : true;
        }
    };

    CPT.getColumnModel().getColumn(0).setCellRenderer(new
CustomRenderer());

    JScrollPane jsp = new JScrollPane();
    jsp.getViewPort().add(CPT);
    int option =
JOptionPane.showConfirmDialog(rootPane, jsp, "CPT "+label,
JOptionPane.OK_CANCEL_OPTION,
JOptionPane.PLAIN_MESSAGE);
    if(option == JOptionPane.OK_OPTION){
        setCPTTable(CPT, label);
    }
}

```

```

private String[] getComb(String node) throws
NeticaException{
    String comb[] = new String[getCPTSize(node).x];
    NodeList temp = net.getNode(node).getParents();
    int flag = 0;
    if(temp.size() == 1){
        int stateP1 = temp.getNode(0).getNumStates();
        for(int j = 0; j < stateP1; j++){

//System.out.println(temp.getNode(0).state(j));
            comb[flag] = temp.getNode(0).state(j).toString();
                flag++;
        }
    }else if(temp.size() == 2){
        int stateP1 = temp.getNode(0).getNumStates();
        int stateP2 = temp.getNode(1).getNumStates();
        for(int i = 0; i < stateP1; i++){
            for(int j = 0; j < stateP2; j++){

//System.out.println(temp.getNode(0).state(i)+","+temp.getN
ode(1).state(j));
                comb[flag] = temp.getNode(0).state(i).toString()+" "+temp.getNode(1).stat
e(j).toString();
                    flag++;
            }
        }
    }
}

```

```

}else if(temp.size() == 3){
    int stateP1 = temp.getNode(0).getNumStates();
    int stateP2 = temp.getNode(1).getNumStates();
    int stateP3 = temp.getNode(2).getNumStates();
    for(int i = 0; i < stateP1; i++){
        for(int j = 0; j < stateP2; j++){
            for(int k = 0; k < stateP3; k++){

                //System.out.println(temp.getNode(0).state(i)+","+temp.getNode(1).state(j)+","+temp.getNode(2).state(k));
                comb[flag] =
temp.getNode(0).state(i).toString()+","+temp.getNode(1).state(j).toString()+","+temp.getNode(2).state(k).toString();
                flag++;
            }
        }
    }
    return comb;
}

private Point getCPTSize(String node) throws NeticaException{
    NodeList temp = net.getNode(node).getParents();
    int col = net.getNode(node).getNumStates();
    int row = 1;
    for(int i = 0; i < temp.size(); i++){

```

```

row *= temp.getNode(i).getNumStates();
}
return new Point(row, col);
}
private String[] getColHeader(int stateSize, String node)
throws NeticaException{
    String[] state = new String[stateSize+1];
    for(int i = 0; i < stateSize+1; i++){
        if(i == 0){
            String parent = "";
            NodeList temp = net.getNode(node).getParents();
            parent += "(";
            for(int s = 0; s < temp.size(); s++){
                if(s != temp.size() - 1){
                    parent += temp.getNode(s).getName()+" | ";
                }else{
                    parent += temp.getNode(s).getName();
                }
            }
            parent += ")";
            state[i] = parent;
        }else{
            state[i] = net.getNode(node).state(i-1).toString();
        }
    }
    return state;
}

```



```

private Object[][] getCPdata(int row, int col, float[] probs,
String[] comb){
    Object[][] data = new Object[row][col+1];
    int k = 0;
    int m = 0;
    for(int i = 0; i < row; i++){
        for(int j = 0; j < col+1; j++){
            if(j == 0){
                data[i][j] = comb[m];
                m++;
            }else{
                data[i][j] = probs[k];
                k++;
            }
        }
    }
    return data;
}

private void setCPTTable(JTable CPT, String label) {
    int row = CPT.getModel().getRowCount();
    int col = CPT.getModel().getColumnCount();
    float[] probs = new float[row*col];
    int k = 0;
    for(int i = 0; i < row; i++){
        for(int j = 1; j < col; j++){
            probs[k] = Float.parseFloat(CPT.getModel().getValueAt(i,
j).toString().replace(", ", "."));

```

```

try {
    net.getNode(label).setCPTable(probs);
} catch (NeticaException ex) {

JOptionPane.showMessageDialog(rootPane,    ex,    "Error
Message", JOptionPane.ERROR_MESSAGE);
    }
}
});
    pop.add(it1);
    pop.show(e.getComponent(), e.getX(), e.getY());
        }
    }

    @Override
    public void mousePressed(MouseEvent e) {
    }

    @Override
    public void mouseReleased(MouseEvent e) {
    }

    @Override
    public void mouseEntered(MouseEvent e) {
    }

```

```

        @Override
        public void mouseExited(MouseEvent e) {
        }
    });

    MouseInteractionSimple mi = new
    MouseInteractionSimple(graphComponent, graph);
    mi.edgeCreationControl();
    mi.zoomControl();
    graphPanel.getViewport().add(graphComponent);
    private void
    exitMenuActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }

    Private void
    saveMenuActionPerformed(java.awt.event.ActionEvent evt) {
        JFileChooser chooser = new JFileChooser("C:\\");
        int retrival = -1;
        NeticaFilter filter = new NeticaFilter();
        chooser.setFileFilter(filter);
        if(net != null){
            try {
                chooser.setSelectedFile(new
                File("C:\\"+net.getName()+".dne"));
                retrival = chooser.showSaveDialog(null);
            } catch (NeticaException ex) {

```

```

Logger.getLogger(BNSimple.class.getName()).log(Level.SEVERE,
null, ex);}
}else{
    JOptionPane.showMessageDialog(rootPane, "Compile
Network First.", "Info",
JOptionPane.INFORMATION_MESSAGE);
}
    if (retrival == JFileChooser.APPROVE_OPTION) {
    try{
System.out.println(chooser.getSelectedFile().getName());
        if(net != null){
            net.write(new
Streamer(chooser.getSelectedFile().getAbsolutePath()+".dne"
));
        }else{
            JOptionPane.showMessageDialog(rootPane, "Compile
Network First.", "Info",
JOptionPane.INFORMATION_MESSAGE);
        }
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(rootPane, ex, "Error",
JOptionPane.ERROR_MESSAGE);
    }
}
}
}

```

```

private void
openMenuActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser chooser = new JFileChooser("C:/");
    //chooser.setFileSelectionMode(JFileChooser.FILES_O
NLY);
    NeticaFilter filter = new NeticaFilter();
    chooser.setFileFilter(filter);
    //int returnVal = chooser.showOpenDialog(null);
    int returnVal = chooser.showOpenDialog(this);
    if(returnVal == JFileChooser.APPROVE_OPTION) {
        newGraph();
        try {
            // TODO add your handling code here:
            net = new Net( new Streamer(
chooser.getSelectedFile().getAbsolutePath() ) );
            buildGraphFromFile();
            net.compile();
            if(net.getTitle().equals("Anchor Damage")){
                resetInfPanel();
                anchorDamageInference ADI = new
anchorDamageInference(net);
                inferencePanel.add(ADI);
                repaintInfPanel();
            }else if(net.getTitle().equals("Trawling Damage")){
                resetInfPanel();
                trawlingDamageInference TDI = new
trawlingDamageInference(net);

```

```

inferencePanel.add(TDI);
    repaintInfPanel();
}
compileBtn.setEnabled(true);
inferenceBtn.setEnabled(true);
} catch (NeticaException ex)
{
    Logger.getLogger(BNSimple.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
private void buildGraphFromFile(){
    try {
        NodeList nodes = net.getNodes();
        for(Object node:nodes){
            double[] position =
net.getNode(node.toString()).visual().getPosition();
            graph.getModel().beginUpdate();
            try{
                graph.insertVertex(parent, null, node.toString(),
                    position[0], position[1],
100,40,"shape=ellipse;perimeter=ellipsePerimeter");
            }
            finally
            {
                graph.getModel().endUpdate();
            }
        }
    }
}

```

```

}
for(Object node:nodes){
    NodeList lp = net.getNode(node.toString()).getParents();
    Object chld = getvertex(node.toString());
    for(Object p:lp){
        Object prnt = getvertex(p.toString());
        graph.getModel().beginUpdate();
        try{
            graph.insertEdge(parent, null, "", prnt, chld);
        }
        finally
        {
            graph.getModel().endUpdate();
        }
    }
}
} catch (NeticaException ex) {
    Logger.getLogger(BNSimple.class.getName()).log(Level.SEVERE, null, ex);
}
}

private Object getvertex(String node){

    for(Object o:graph.getChildVertices(parent)){
        if(graph.getLabel(o).equals(node)){
            return o;
        }
    }
}
}

```

```

private void
compileBtnActionPerformed(java.awt.event.ActionEvent evt)
{
    try {
        // TODO add your handling code here:
        net.compile();
        String title = net.getTitle();
        NetPanel netPanel = new NetPanel(net,
NodePanel.NODE_STYLE_BELIEF_BARS);
        netPanel.addListenerToAllComponents(new
MyMouseListener(netPanel));
        JScrollPane jsp = new JScrollPane(netPanel);
        JOptionPane.showMessageDialog(rootPane, jsp, title,
JOptionPane.PLAIN_MESSAGE);
        //net.finalize();
        //netPanel.addListenerToAllComponents( new
MyMouseListener(netPanel) );
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(rootPane,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}
private void
chooseBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    compileBtn.setEnabled(true);
    inferenceBtn.setEnabled(true);
    if(hazard1.isSelected()){
        generateHazardnNet("hazard1");
    } else if(hazard2.isSelected()){

```



```

generateHazardnNet("hazard2");
    }
}
private void generateHazardnNet(String type){
    generateHazard(type);
    generateNet(type);
}
Private                                     void
hazard2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    disableCIBtn();
    if(hazard2.isSelected()){
        chooseBtn.setEnabled(true);
    }
}
private                                     void
hazard1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    disableCIBtn();
    if(hazard1.isSelected()){
        chooseBtn.setEnabled(true);
    }
}
private void disableCIBtn(){
    compileBtn.setEnabled(false);
    inferenceBtn.setEnabled(false);
}

```

```

private void
inferenceBtnActionPerformed(java.awt.event.ActionEvent
evt) {
    try {
        // TODO add your handling code here:
        net.compile();
        if(net.getTitle().equals("Anchor Damage")){
            resetInfPanel();
            anchorDamageInference      ADI      =      new
anchorDamageInference(net);
            inferencePanel.add(ADI);
            repaintInfPanel();
        }else if(net.getTitle().equals("Trawling Damage")){
            resetInfPanel();
            trawlingDamageInference      TDI      =      new
trawlingDamageInference(net);
            inferencePanel.add(TDI);
            repaintInfPanel();
        }
    } catch (NeticaException ex) {

        Logger.getLogger(BNSimple.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

private void resetInfPanel(){
    inferencePanel.removeAll();
    inferencePanel.setLayout(new GridLayout());
}

```

```

private void repaintInfPanel(){
    inferencePanel.repaint();
    inferencePanel.revalidate();
    jTabbedPane1.setEnabledAt(1, true);
    jTabbedPane1.setSelectedIndex(1);
}
private void generateHazard(String hazard){
    newGraph();
    Random rand = new Random();
    String[] vertex; Point[] point; Point[] edgePoint;
    if(hazard.equals("hazard1")){
        anchorDamage ad = new anchorDamage();
        vertex = ad.getVertex();
        point = ad.getPoint();
        edgePoint = ad.getEdgePoint();
    }else{
        trawlingDamage td = new trawlingDamage();
        vertex = td.getVertex();
        point = td.getPoint();
        edgePoint = td.getEdgePoint();
    }
    buildGraph(vertex, point, edgePoint);
}
private void buildGraph(String[] vertex, Point[] point, Point[]
edgePoint){
    graph.getModel().beginUpdate();
    try
    {
        for(int i = 0; i < vertex.length; i++){

```

```

graph.insertVertex(parent, null, vertex[i], point[i].getX(),
point[i].getY(),
100,40,"shape=ellipse;perimeter=ellipsePerimeter");
    }
    for(int i = 0; i < edgePoint.length; i++){
        graph.insertEdge(parent, null, "",
graph.getModel().getChildAt(parent, edgePoint[i].x),
graph.getModel().getChildAt(parent, edgePoint[i].y));
    }
}
finally
{
    graph.getModel().endUpdate();
}
}

private void generateNet(String hazard){
    try {
        String[] vertex = null; Point[] point = null; Point[]
edgePoint = null; String[] state = null;
Node.setConstructorClass("norsys.neticaEx.aliases.Node");
        net = new Net();
        net.setName("BayesianNetwork");
        if(hazard.equals("hazard1")){
            net.setTitle("Anchor Damage");
            anchorDamage ad = new anchorDamage();
            vertex = ad.getVertex();
            point = ad.getPoint2();
            edgePoint = ad.getEdgePoint();
            state = ad.getState();

```

```

}else if(hazard.equals("hazard2")){
    net.setTitle("Trawling Damage");
    trawlingDamage td = new trawlingDamage();
    vertex = td.getVertex();
    point = td.getPoint2();
    edgePoint = td.getEdgePoint();
    state = td.getState();
}
buildNet(vertex, state, point, edgePoint);

} catch (Exception e) {
    e.printStackTrace();
}
}

private void buildNet(String[] vertex, String[] state, Point[]
point, Point[] edgePoint) throws Exception{
    for(int i = 0; i < vertex.length; i++){
        new Node (vertex[i], state[i], net);

net.getNode(vertex[i]).visual().setPosition(point[i].getX(),
point[i].getY());
    }
    for(int i = 0; i < edgePoint.length; i++){
        net.getNode(vertex[edgePoint[i].y]).addLink(
net.getNode(vertex[edgePoint[i].x]));
    }
    setInitialCPT(net);
}

```

```

private void setCPT(String node, int row, int col) throws
NeticaException{
    int stateNum = net.getNode(node).getNumStates();
    float[] probs = new float[row*col];
    if(node.equals("engineering_ship")           ||
node.equals("transport_ship")                   ||
node.equals("fishing_vessel")){
        for(int i = 0; i < row*col; i++){
            probs[i] = (float) (1.0/stateNum);
        }
    }else{
        for(int i = 0; i < row*col; i++){
            if(i%stateNum == 0){
                probs[i] = 1.0F;
            }else{
                probs[i] = 0.0F;
            }
        }
    }
    net.getNode(node).setCPTTable(probs);
}

private void newGraph(){
    graphPanel.getViewport().removeAll();
    graphPanel.repaint();
    graphPanel.revalidate();
}

private void newGraph(){
    graphPanel.getViewport().removeAll();
    graphPanel.repaint();
    graphPanel.revalidate();
}

```

```

graph.removeCells(graph.getChildVertices(graph.getDefaultParent()));
    graphComponent.updateComponents();
    graphPanel.getViewport().add(graphComponent);
}
public static <V,E> void removeAllEdges(Graph<V, E> graph)
{
    LinkedList<E> copy = new LinkedList<E>();
    for (E e : graph.edgeSet()) {
        copy.add(e);
    }
    graph.removeAllEdges(copy);
}
public static <V,E> void clearGraph(Graph<V, E> graph) {
    removeAllEdges(graph);
    removeAllVertices(graph);
}
public static <V,E> void removeAllVertices(Graph<V, E> graph)
{
    LinkedList<V> copy = new LinkedList<V>();
    for (V v : graph.vertexSet()) { copy.add(v);}
    graph.removeAllVertices(copy);
}
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new BNSimple().setVisible(true);
        }
    });
}

```

## 2. Mengubah Nilai Node Menjadi Vektor (*Coloumn Grup*)

```

class ColumnGroup {
    protected TableCellRenderer renderer;
    protected Vector v;
    protected String text;
    protected int margin=0;

    public ColumnGroup(String text) {
        this(null,text) }
    public ColumnGroup(TableCellRenderer renderer,String
text) {
        if (renderer == null) {
            this.renderer = new DefaultTableCellRenderer() {
                public Component
getTableCellRendererComponent(JTable table, Object value,
boolean isSelected, boolean hasFocus, int row, int column)
JTableHeader header = table.getTableHeader();
                if (header != null) {
                    setForeground(header.getForeground());
                    setBackground(header.getBackground());
                    setFont(header.getFont());
                }
            }
        setHorizontalAlignment(JLabel.CENTER);
        setText((value == null) ? "" : value.toString());

        setBorder(UIManager.getBorder("TableHeader.cellBorder"))
    ;
    return this; } };

```



```

} else {
    this.renderer = renderer;
}
this.text = text;
v = new Vector();
}
public void add(Object obj) {
    if (obj == null) { return; }
    v.addElement(obj);
}
public Vector getColumnGroups(TableColumn c, Vector g) {
    g.addElement(this);
    if (v.contains(c)) return g;
    Enumeration e = v.elements();
    while (e.hasMoreElements()) {
        Object obj = e.nextElement();
        if (obj instanceof ColumnGroup) {
            Vector groups =
(Vector)((ColumnGroup)obj).getColumnGroups(c,(Vector)g.clone());
            if (groups != null) return groups;
        }
    }
    return null;
}
public TableCellRenderer getHeaderRenderer() {
    return renderer;
}

```

```

public void setHeaderRenderer(TableCellRenderer renderer) {
    if (renderer != null) {
        this.renderer = renderer;
    }
}

public Object getHeaderValue() {
    return text;
}

public Dimension getSize(JTable table) {
    Component comp =
renderer.getTableCellRendererComponent(
    table, getHeaderValue(), false, false, -1, -1);
    int height = comp.getPreferredSize().height;
    int width = 0;
    Enumeration e = v.elements();
    while (e.hasMoreElements()) {
        Object obj = e.nextElement();
        if (obj instanceof TableColumn) {
            TableColumn aColumn = (TableColumn)obj;
            width += aColumn.getWidth();
            width += margin;
        } else {
            width += ((ColumnGroup)obj).getSize(table).width;
        }
    }
    return new Dimension(width, height);
}

```

```

public void setColumnMargin(int margin) {
    this.margin = margin;
    Enumeration e = v.elements();
    while (e.hasMoreElements()) {
        Object obj = e.nextElement();
        if (obj instanceof ColumnGroup) {
            ((ColumnGroup)obj).setColumnMargin(margin); }
        } }
}

```

### 3. Mengubah Nilai Node Menjadi Vektor (*Custom Render*)

```

public class CustomRenderer extends
DefaultTableCellRenderer {
    private static final long serialVersionUID =
6703872492730589499L;

    public Component
getTableCellRendererComponent(JTable table, Object value,
boolean isSelected, boolean hasFocus, int row, int column)
{
    Component cellComponent =
super.getTableCellRendererComponent(table, value,
isSelected, hasFocus, row, column);
    if(column == 0){
        cellComponent.setBackground(Color.YELLOW);
        cellComponent.setForeground(Color.BLACK);
    }
    return cellComponent; }
}

```

#### 4. *Mouse Input Adapter*

```

public class MyMouseListener extends
MouseListener {
    Point pressPt;
    boolean dragged;
    static NetPanel curNetPanel = null;

    public MyMouseListener(NetPanel curNetPanel){
        this.curNetPanel = curNetPanel;
    }
    public void mouseClicked(MouseEvent me) {
        Component comp = me.getComponent();
        //System.out.println( "Mouse clicked on " +
comp.getClass().getName() );
        try {
            if (comp instanceof JLabel) comp =
comp.getParent(); //pass on to containing NetPanel
            if (comp instanceof NetPanel ) {
                NetPanel np = (NetPanel) comp;
                //System.out.println( "Mouse clicked on NetPanel
for node " + np.getNode().getName() );
                if ( me.paramString().indexOf( "button=1" ) >= 0 ) {
//-- left button was clicked
                    if(np.getDisplayMode()==
NetPanel.DISPLAY_MODE_NORMAL ) {
                        np.setDisplayMode(
NetPanel.DISPLAY_MODE_HILITED );
                    }
                }
            }
        }
    }
}

```

```

else if
(np.getDisplayMode() == NodePanel.DISPLAY_MODE_HILITED
){
np.setDisplayMode( NodePanel.DISPLAY_MODE_NORMAL );
    }
}else
if ( (me.paramString().indexOf( "button=2" ) >= 0 ) ||
(me.paramString().indexOf( "button=3" ) >= 0 ) ){ //-- middle
or right button was clicked
//... handle right/middle button click as desired; for now, just
log event to console
//System.out.println( "Right  mouse  clicked  on  "  +
comp.getClass().getName() );
    }
}
else if (comp instanceof NodePanel_BeliefBarsRow) {
    NodePanel_BeliefBarsRow      npbbr      =
(NodePanel_BeliefBarsRow) comp;
    int indexOfStateClicked = npbbr.getState().getIndex();
    Value finding = npbbr.getState().getNode().finding();
    if ( finding.getState() == indexOfStateClicked) {
        //-- clicking on already selected state
        finding.clear();
    }
    else {
        finding.setState( indexOfStateClicked );}
    try{
        curNetPanel.getNet().compile();

```

```

else if (comp instanceof LinkGraphic) {
    LinkGraphic link = (LinkGraphic) comp;
    System.out.println( "Mouse clicked on link: " + link +
", at point " + me.getPoint() );
    if(link.getDisplayMode()==
LinkGraphic.DISPLAY_MODE_NORMAL ) {
        link.setDisplayMode(
LinkGraphic.DISPLAY_MODE_HILITED );
//link.setLineWidth( 3 ); // basic stroke, which will work fine,
but...
// ... you can go wild too!!
float[] dash = {10.0F, 5.0F};
link.setStroke(      new      BasicStroke(      3.0F,
BasicStroke.CAP_ROUND,   BasicStroke.JOIN_ROUND,   1.0F,
dash, 0.0F ) );
//just for fun, if it is Button #2 or #3, move the link's endpoint
    if ( me.getButton() != MouseEvent.BUTTON1 ) {
        Point start = link.getStartPoint();
        link.setEndPoint( me.getPoint() );
        }
        curNetPanel.repaint();
    }
    else {
        link.setDisplayMode(
LinkGraphic.DISPLAY_MODE_NORMAL );
        link.setStroke( new BasicStroke( 1.0F ) );
        curNetPanel.refreshDataDisplayed(); //NB this resets Link
        coords to match underlying Nodes
    }
}

```

```

    }
    }
    catch (Exception e2) {
        JOptionPane.showMessageDialog(comp,
e2.getMessage() );
        //e2.printStackTrace(); //TBD
    }
}

```

## 5. Netica Filter

```

public class NeticaFilter extends
javax.swing.filechooser.FileFilter{
    String description
    //Accept all directories and all neta, dne, and dnet extension
    files
    public boolean accept(File f) {
        if (f.isDirectory()) {
            return true;
        }
        String extension = getExtension(f.getName());
        if (extension != null) {
            if (extension.equals("neta") ||
                extension.equals("dne") ||
                extension.equals("dnet") ) {
                return true;
            } else {
                return false;
            }
        }
    }
}

```

```

public String getDescription() {
    return ".dne, .dnet, .neta : Netica format nets";
}
public String getExtension(String fileName) {
    int pos = fileName.lastIndexOf('.');
    if (pos<0)
        return "";
    else
        return fileName.substring(pos+1);
}
}

```

## 6. *Anchor Damage*

```

public class anchorDamage {
    private final String[] vertex;
    private final String[] state;
    private final Point[] point;
    private final Point[] point2;
    private final Point[] edgePoint;

    public anchorDamage(){
        this.vertex = generateData();
        this.point = generatePoint();
        this.point2 = generatePoint2();
        this.edgePoint = generateEdgePoint();
        this.state = generateState();
    }
}

```



```

private String[] generateData(){
    String[] v = {"engineering_ship", "transport_ship",
        "fishing_vessel", "engine" "ship_passing",
        "anchor_weight", "buried_depth", "anchor", "water_depth",
        "impact", "impact_energy", "anchor_damage"};
        return v;
    }
    private String[] generateState() {
        String[] s = {"passing, no", "passing, no", "passing, no", "fail,
            work", "passing, no", "no, light, medium, heavy", "shallow,
            medium, deep", "anchor, no", "shallow, medium, deep",
            "impact,no","no,low,medium,high,tremendous","no,minor,m
            oderate,major"};
            return s;
        }
        private Point[] generatePoint(){
            Point[] p = {new Point(100, 10),
                new Point(300, 10),
                new Point(500, 10),
                new Point(50, 80),
                new Point(200, 80),
                new Point(400, 80),
                new Point(600, 80),
                new Point(100, 150),
                new Point(300, 150),
                new Point(200, 250),
                new Point(400, 250),
                new Point(300, 350),
            };
        }
    }

```

```
return p;
}

private Point[] generatePoint2(){
    Point[] p = {new Point(150, 10),
        new Point(350, 10),
        new Point(550, 10),
        new Point(100, 150),
        new Point(300, 150),
        new Point(500, 150),
        new Point(700, 150),
        new Point(200, 250),
        new Point(400, 250),
        new Point(300, 350),
        new Point(500, 350),
        new Point(400, 500),
    };
    return p;
}

private Point[] generateEdgePoint(){
    Point[] ep = {new Point(0, 5),
        new Point(0, 4),
        new Point(1, 4),
        new Point(1, 5),
        new Point(2, 4),
        new Point(2, 5),
        new Point(3, 7),
        new Point(4, 7),
        new Point(5, 10),
        new Point(6, 10),
```

```
        new Point(7, 9),
            new Point(8, 9),
            new Point(10, 11),
            new Point(9, 11),
        };
        return ep;
    }
    public String[] getVertex() {
        return vertex;
    }
    public Point[] getPoint() {
        return point;
    }
    public Point[] getEdgePoint() {
        return edgePoint;
    }
    public String[] getState() {
        return state;
    }
    public Point[] getPoint2() {
        return point2;
    }
}
```

## 9. Inferensi Anchor Damage

```

public      class      anchorDamageInference      extends
javax.swing.JPanel {
    DecimalFormat fmt;
    Net net;
    public anchorDamageInference(){
        initComponents(); }
    public      anchorDamageInference(Net      net)      throws
NeticaException {
        initComponents();
        this.fmt = new DecimalFormat( "###.###" );
        fmt.setMinimumFractionDigits(2);
        this.net = net;
        setActionCommand();
        initBtn();
        doInference() }
private      void      setActionCommand(){
    espass.setActionCommand("passing");esno.setActionComma
nd("no");esdkn.setActionCommand("dkn");
    tspass.setActionCommand("passing");tsno.setActionComman
d("no");tsdkn.setActionCommand("dkn");
    fvpas.setActionCommand("passing");fvno.setActionComman
d("no");fvdkn.setActionCommand("dkn");
    effail.setActionCommand("fail");efwork.setActionCommand("
work");efdkn.setActionCommand("dkn");
    wdshallow.setActionCommand("shallow");wdmedium.setActi
onCommand("medium");wddeep.setActionCommand("deep")
;wddkn.setActionCommand("dkn");

```

```

pbdshallow.setActionCommand("shallow");pbdmedium.setAc
tionCommand("medium");pbddeep.setActionCommand("dee
p");pbddkn.setActionCommand("dkn");
    }
private void initBtn(){
    esdkn.setSelected(true);
    tsdkn.setSelected(true);
    fvdkn.setSelected(true);
    efdkn.setSelected(true);
    wddkn.setSelected(true);
    pbddkn.setSelected(true);
    }
private void submitBtnActionPerformed(java.awt.event.ActionEvent evt) {
    doInference();}
private void doInference(){
    try {
        net.retractFindings();
        shipPassingProb();
        anchorWeightProb();
        impactEnergyProb();
        anchorProb();
        impactProb();
        anchorDamageProb();
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}
}

```

```

private void shipPassingProb(){
    try {
        String          engineerShip          =
es_state.getSelection().getActionCommand();
        String          transportShip          =
ts_state.getSelection().getActionCommand();
        String          fishingVesel          =
fv_state.getSelection().getActionCommand();
        if(engineerShip != "dkn"){

net.getNode("engineering_ship").finding().enterState(enginee
rShip);
        }
        if(transportShip != "dkn"){

net.getNode("transport_ship").finding().enterState(transportS
hip);
        }
        if(fishingVesel != "dkn"){
net.getNode("fishing_vessel").finding().enterState(fishingVese
l);}

        sp.setText("Passing                                :
"+fmt.format(net.getNode("ship_passing").getBelief("passing
")));
        snp.setText("Not                                passing                                :
"+fmt.format(net.getNode("ship_passing").getBelief("no")));
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);}

```

```

private void anchorProb(){
    try {
        String engine =
ef_state.getSelection().getActionCommand();
        if(engine != "dkn"){
            net.getNode("engine").finding().enterState(engine);
        }
    }
    ap.setText("Anchoring :
"+fmt.format(net.getNode("anchor").getBelief("anchor")));
    nap.setText("Not anchoring :
"+fmt.format(net.getNode("anchor").getBelief("no")));
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void anchorWeightProb(){
    try {
        awn.setText("No anchor :
"+fmt.format(net.getNode("anchor_weight").getBelief("no")))
;
        awl.setText("Light anchor :
"+fmt.format(net.getNode("anchor_weight").getBelief("light"
)));
        awm.setText("Medium anchor :
"+fmt.format(net.getNode("anchor_weight").getBelief("medi
um")));
        awh.setText("Heavy anchor :
"+fmt.format(net.getNode("anchor_weight").getBelief("heavy
")));
    }
}

```

```

awh.setText("Heavy                                anchor                                :
"+fmt.format(net.getNode("anchor_weight").getBelief("heavy
")));
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}
private void impactProb(){
    try {
        String                                waterDepth                                =
wd_state.getSelection().getActionCommand();
        if(waterDepth                                !=                                "dkn"){
net.getNode("water_depth").finding().enterState(waterDepth
);
        }
        ipi.setText("Impact                                :
"+fmt.format(net.getNode("impact").getBelief("impact")));
        ipn.setText("No                                :
"+fmt.format(net.getNode("impact").getBelief("no")));
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    } }
private void impactEnergyProb(){
    try {
        String                                buriedDepth                                =
bd_state.getSelection().getActionCommand();
        if(buriedDepth != "dkn"){

```



```

        ien.setText("0                kJ                :
"+fmt.format(net.getNode("impact_energy").getBelief("no")));
        iel.setText("<                2.5                kJ                :
"+fmt.format(net.getNode("impact_energy").getBelief("low")))
    );
        iem.setText("2.5                -                10                kJ                :
"+fmt.format(net.getNode("impact_energy").getBelief("medi
um"))));
        ieh.setText("10                -                20                kJ                :
"+fmt.format(net.getNode("impact_energy").getBelief("high"
))));
        iet.setText(">                20                kJ                :
"+fmt.format(net.getNode("impact_energy").getBelief("treme
ndous"))));
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void anchorDamageProb(){
    try {
        adn.setText("No                :
"+fmt.format(net.getNode("anchor_damage").getBelief("no"
))));
        admn.setText("Minor                :
"+fmt.format(net.getNode("anchor_damage").getBelief("min
or"))));
        admd.setText("Moderate                :
"+fmt.format(net.getNode("anchor_damage").getBelief("mod
erate"))));
    }
}

```

```

admj.setText("Major                                     :
"+fmt.format(net.getNode("anchor_damage").getBelief("maj
or")));
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}
private                                     void
resetBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    initBtn();
    doInference();
}

```

## 10. *Trawling Damage*

```

public class trawlingDamage {
    private final String[] vertex;
    private final String[] state;
    private final Point[] point;
    private final Point[] point2;
    private final Point[] edgePoint;

    public trawlingDamage(){
        this.vertex = generateData();
        this.point = generatePoint();
        this.point2 = generatePoint2();
        this.edgePoint = generateEdgePoint();
        this.state = generateState();
    }

    private String[] generateData(){
        String[] v = {"fishing_vessel", "casting_net", "max_depth",
"water_depth", "bottom_trawling", "buried_depth",
        "hooking", "drow_energy", "trawling_damage"};
        return v;
    }

    private String[] generateState() {
        String[] s = {"passing, no","casting, no", "yes, no",
"shallow, medium, deep", "yes, no", "shallow, medium,
deep",      "yes, no", "no, low, medium, high, tremendous",
"no, minor, moderate, major"};
        return s;
    }
}

```

```
private Point[] generatePoint(){
    Point[] p = {new Point(350, 10),
        new Point(200, 50),
        new Point(100, 100),
        new Point(300, 100),
        new Point(200, 150),
        new Point(400, 150),
        new Point(300, 200),
        new Point(500, 200),
        new Point(400, 250)
    };
    return p;
}

private Point[] generatePoint2(){
    Point[] p = {new Point(350, 10),
        new Point(200, 100),
        new Point(100, 200),
        new Point(400, 200),
        new Point(200, 300),
        new Point(500, 300),
        new Point(300, 400),
        new Point(550, 400),
        new Point(420, 530)
    };
    return p;
}
```

```
private Point[] generateEdgePoint(){
    Point[] p = {new Point(0, 1),
        new Point(1, 2),
        new Point(3, 2),
        new Point(3, 4),
        new Point(2, 4),
        new Point(4, 6),
        new Point(5, 6),
        new Point(6, 7),
        new Point(6, 8),
        new Point(7, 8),
    };
    return p;
}
public String[] getVertex() {
    return vertex;
}
public Point[] getPoint() {
    return point;
}
public Point[] getEdgePoint() {
    return edgePoint;
}
public String[] getState() {
    return state;
}
public Point[] getPoint2() {
    return point2;
} }
```

11. Inferensi *Trawling Damage*

```

public class trawlingDamageInference extends
javax.swing.JPanel {
    Net net;
    DecimalFormat fmt;
    public trawlingDamageInference() {
        initComponents();
    }
    public trawlingDamageInference(Net net) {
        initComponents();
        this.fmt = new DecimalFormat( "###.###" );
        fmt.setMinimumFractionDigits(2);
        this.net = net;
        addToGroupBtn();
        setActionCommand();
        initBtn();
        doInference();
    }
    private void addToGroupBtn(){

        fv_state.add(fvpass);fv_state.add(fvno);fv_state.add(fvdkn);

        wd_state.add(wdshallow);wd_state.add(wdmedium);wd_state.add(wddeep);wd_state.add(wddkn);

        bd_state.add(bdshallow);bd_state.add(bdmedium);bd_state.add(bddeep);bd_state.add(bddkn);
    }
}

```

```

private void setActionCommand(){
    fvpas.setActionCommand("passing");fvno.setActionCommand("no");fvdkn.setActionCommand("dkn");
    wdshallow.setActionCommand("shallow");wdmedium.setActionCommand("medium");wddeep.setActionCommand("deep");
    wddkn.setActionCommand("dkn");
    bdshallow.setActionCommand("shallow");bdmedium.setActionCommand("medium");bddeep.setActionCommand("deep");
    bddkn.setActionCommand("dkn");
}
private void initBtn(){
    fvdkn.setSelected(true);
    wddkn.setSelected(true);
    bddkn.setSelected(true);
}
private void doInference(){
    try {
        net.retractFindings();
        shipPassingProb();
        maxDepthProb();
        bottomTrawlingProb();
        hookingProb();
        drowEnergyProb();
        trawlingDamageProb();
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
        ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}
}

```

```

private void shipPassingProb(){
    try {
        String fishingVesel =
fv_state.getSelection().getActionCommand();
        if(fishingVesel != "dkn"){
net.getNode("fishing_vessel").finding().enterState(fishingVese
l);
        }
        cny.setText("Casting Net :
"+fmt.format(net.getNode("casting_net").getBelief("casting")
));
        cnn.setText("Not Casting Net :
"+fmt.format(net.getNode("casting_net").getBelief("no")));
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}
private void maxDepthProb(){
    try {
        String waterDepth =
wd_state.getSelection().getActionCommand();
        if(waterDepth != "dkn"){
net.getNode("water_depth").finding().enterState(waterDepth
);
        }
        mdy.setText("Max Depth :
"+fmt.format(net.getNode("max_depth").getBelief("yes")));
        mdn.setText("No :
"+fmt.format(net.getNode("max_depth").getBelief("no")));
    }
}

```



```

    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void bottomTrawlingProb(){
    try {
        btn.setText("Bottom Trawling :
"+fmt.format(net.getNode("bottom_trawling").getBelief("yes
")));
        btn.setText("No :
"+fmt.format(net.getNode("bottom_trawling").getBelief("no
")));
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void hookingProb(){
    try {
        String buriedDepth =
bd_state.getSelection().getActionCommand();
        if(buriedDepth != "dkn"){
net.getNode("buried_depth").finding().enterState(buriedDept
h);
        }
    }
}

```

```

}
    hy.setText("Hooking                                :
"+fmt.format(net.getNode("hooking").getBelief("yes")));
    hn.setText("No                                    :
"+fmt.format(net.getNode("hooking").getBelief("no")));
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}
private void drawEnergyProb(){
    try {
        drowno.setText("No                                Energy:
"+fmt.format(net.getNode("drow_energy").getBelief("no")));
        drowlow.setText("Low                                :
"+fmt.format(net.getNode("drow_energy").getBelief("low")));
        drowmedium.setText("Medium                                :
"+fmt.format(net.getNode("drow_energy").getBelief("mediu
m")));
        drowhigh.setText("High                                :
"+fmt.format(net.getNode("drow_energy").getBelief("high")))
;
        drowtre.setText("Tremendous                                :
"+fmt.format(net.getNode("drow_energy").getBelief("tremen
dous")));
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);    }

```

```

private void trawlingDamageProb(){
    try {
        tdn.setText("No Damage :
"+fmt.format(net.getNode("trawling_damage").getBelief("no
")));
        tdmn.setText("Minor Damage :
"+fmt.format(net.getNode("trawling_damage").getBelief("mi
nor")));
        tdmnd.setText("Moderate Damage :
"+fmt.format(net.getNode("trawling_damage").getBelief("mo
derate")));
        tdmj.setText("Major Damage :
"+fmt.format(net.getNode("trawling_damage").getBelief("ma
jor")));
    } catch (NeticaException ex) {
        JOptionPane.showMessageDialog(this,
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void
submitBtnActionPerformed(java.awt.event.ActionEvent evt) {
    doInference();
}

```



## **LAMPIRAN C**

### **BIODATA PENULIS**



Penulis bernama Firda Puspita Devi atau yang akrab disapa Firda, lahir di Madiun, 5 Februari 1995. Penulis merupakan anak pertama pasangan Mahfudz Sayudi dan Novaria Tuesdiana. Penulis menempuh pendidikan formal di SD Latihan YBBSU Balikpapan (2000-2006), SMP Negeri 35 Surabaya (2006-2009), SMA Negeri 16 Surabaya (2009-2012).

Setelah lulus dari SMA, penulis melanjutkan studi ke jenjang S1 di Jurusan Matematika ITS Surabaya pada tahun 2012. Penulis yang mempunyai hobi berpergian ini, kemudian memutuskan untuk melanjutkan studi ke jenjang S1 Jurusan Akuntansi STIE Urip Sumoharjo Surabaya pada tahun 2013. Di Jurusan Matematika, penulis mengambil Bidang Minat Ilmu Komputer. Selain aktif kuliah, penulis juga aktif mengikuti organisasi, di antaranya: staff Hubungan Luar HIMATIKA ITS (2013-2014), Kepala Divisi *literature* IFLS ITS (2014-2015), dan anggota Yosakoi ITS (2014-2015). Tidak hanya itu, Penulis juga aktif berperan dalam beberapa kegiatan kepanitiaan, seperti: Olimpiade Matematika ITS (2013-2015), LKMM Pra TD (2014), LKMM TD (2014) dan ITS Expo (2014). Adapun saran, kritik, dan pertanyaan mengenai Tugas Akhir ini, dapat ditujukan melalui email penulis: [firdapdevi@gmail.com](mailto:firdapdevi@gmail.com).